콤퓨터보안

리 과 대 학
외국문도서출판사
주체91

콤퓨터보안

리과대학
외국문도서출판사

차 례

제2편. 실천

제3편. 분산체계

머 리 말

이 책은 정보보안을 전공하는 박사원생들에게 1년동안 배워 준 강의록을 기초로 하여 만들어 졌다. 이 책은 콤퓨터보안에 대한 편람이나 백과전서가 아니며 콤퓨터보안의 력사를 서술한 책도 아니다. 물론 이 책은 콤퓨터보안력사에서 있었던 사건들에 대하여 언급하고 있으며 보안기술자들이 유익한 착상과 내용을 찾아 볼수 있도록 이 분야의 배경과 보충자료들을 제시하고 있다. 그렇지만 이 책은 무엇보다도 콤퓨터보안에 관한 교과서인것만큼 콤퓨터과학에 대한 일정한 지식을 가지고 있는 학생들로 하여금 여러가지 보안제품들이 가지고 있는 기술적특징들을 대비하고 평가할수 있는 능력을 키워 주는것을 주되는 목적으로 삼고 있다.

이 책에서 기술적측면에 집중하려는 결심은 의도적인것이다. 여기서는 일반정보보안이 아니라 콤퓨터보안이 주요내용을 이루고 있다. 그러므로 위험분석이나 보안관리와 같은 문제들이 언급되여 있지 않다고 하여 그것들이 콤퓨터보안에서는 중요하지 않은것으로 잘못 인식하지 말아야 한다. 그와 반대로 기술적보안문제에 대한 결심은 우의 문제들이 적용되는 정황에 대한 참조없이 이루어 질수 없다. 따라서 기술적문제에 대한 과목들은 콤퓨터보안을 둘러 싸는 비기술적보안문제를 포괄하는 보안관리에 대한 과목들과 나란히 가르쳐야 한다. 기술적문제들만을 가지고서는 체계가 안전하다고 말할수 없을것이다.

이 책은 4개의 편으로 구성되여 있다. 첫편은 5개의 장, 나머지 3개의 편은 4개의 장으로 되여 있다. 4개의 편은 다음과 같다.

제1편. 기초: 여기서는 콤퓨터보안의 기초개념들과 콤퓨터체계의 중심에 놓이면서 다른 보안기구들의 기초를 이루는 기구들이 제시된다.

제2편. 실천: Unix와 Windows NT조작체계의 두가지 경우의 구체적실례를 통하여 여기에서 발로된 결함들을 묶어서 제시하며 보안평가에 대한 륜곽을 보여 준다.

제3편. 분산체계: Web보안을 비롯한 콤퓨터망과 관련된 문제들을 취급한다. 이러

한 환경에서는 암호화가 보안기구들의 기본수단으로 된다. 이 편은 《콤퓨터보안의 최근동향》이라고도 할수 있을것이다.

제4편. 리론: 이 편은 자료기지보안을 위주로 구성되여 있다. 여기서 취급하는 여러준위보안에는 보안문제들을 엄밀하게 검토할수 있는 리론들이 개척되여 있다.

사실 이 책의 거의 모든 장들은 옹근 하나의 책으로 될수 있는 내용들이지만 여기서는 필요한 내용들만을 한개 장의 범위에서 취급하였다. 이 책의 내용은 전반적으로 상당히 포괄적으로 되여 있으며 인쇄물이나 전자매체로 된 보충적인 안내도서들을 지적하여주고 있다. 여기서 쓸모 있는 자료들이 들어 있는 Web싸이트들을 주고 있는데 그 자료들이 다른 싸이트에 옮겨 지지 않았다는 담보는 없으며 따라서 독자가 거기에 접근하려고 할 때에는 그 자료가 이미 쓰일수 없게 될수도 있다. 여기서 선택된 싸이트들이 충분히 안정하고 독자들에게 실망을 주지 않기를 바란다.

이 책을 보는데서 다음과 같은 두가지 문제를 주의해야 한다.

- 제4장의 《보안모형들》보다 먼저 제3장 《접근조종구조》를 주었는데 접근조종을 위한 방책들을 언급하지 않고 접근조종기구들을 취급한것이다. 그러나 어떤 특정한 방책에 의존하지 않는 구조들을 론하려고 한다.
- 제12장의 《암호화》보다 먼저 제10장 《분산체계보안》을 주었는데 이것은 분산체계보안에서 쓰이는 암호화에 대한 기초를 알지 못한 상태에서 분산체계보안을 대하게 된다는것이다. 그러나 독자들은 암호화알고리듬을 그의 거동만이 추상적으로 묘사된 《검은통》으로 취급하면서 보안규칙들을 자체로 해석하기 위해 노력하기 바란다.

이 책의 자료들을 강의에 리용할 때 다음과 같은 점들을 고려하여야 한다.

- 실천을 목적으로 하는 강의에서는 제4장의 보안모형들과 제5장에 있는 특정한 체계들의 기술적상세는 간단히 스치면서 1장부터 14장까지의 자료들을 취급할수 있다.
- 콤퓨터망보안을 다른 과목에서 따로 취급하는 경우에는 제3편의 자료들은 략하고 제4편의 자료들을 취급하기 바란다.

- 강의에서 리론을 중심에 둔다면 사례연구는 략하고 많은 시간을 제4편의 자료들을 취급하는데 돌리기 바란다.

매개 장에 련습문제들을 주고 있는데 통과암호보안이나 암호화와 같이 련습문제가 명백한 답을 얻을수 있는 부분을 제외하고는 토론이나 소론문 같은 형식으로 배운 내용을 익히도록 하였다.

이 책을 통하여 독자들이 현실에서 쓸모 있는 지식을 소유하며 새로운 보안문제들을 발견해 내기를 바란다.

제1편. 기 초

제1장. 일반개념

이 장의 첫번째 과제는 《콤퓨터보안의 정의》에 대한 연구이다. 여기서는 외계와 격리되여 있는 개별적인 보안체계들에 대하여서는 론의하지 않고 보안정보처리체계를 설계하기 위한 일반적인 공학적원리들을 제안한다.

목적

- 기밀성(Confidentiality), 완정성(Integrity), 리용성(Availability)을 도입하여 콤퓨터보안의 정의에 착수한다.
- 콤퓨터보안의 기본난점을 설명한다.
- 보안체계를 구축할 때 채택하여야 할 몇가지 일반적인 설계결심들을 언급한다.
- 콤퓨터보안기구는 물리적인 또는 조직적인 보호기구들에 의존하여야 효과적이라는것을 지적한다.

제1절. 정의

우리의 연구를 학습대상의 정의로부터 시작하자. 콤퓨터보안은 콤퓨터체계내에서 보안을 보장하기 위한 기술들을 취급한다. 여기서는 콤퓨터체계(처리기와 기억기가 들어있는 함)와 정보기술(IT)체계(콤퓨터체계들이 밀접히 결합된 망)를 구별하지 않는다. 기술은 매우 빨리 발전하고 있다. 현대콤퓨터들은 이미 여러 요소들이 밀접히 결합된 하나의 망과 같다. 한때 하나의 응용프로그람이였던 쏘프트웨어들이 조작체계의 한부분으로 될수도 있다. 이와 같은 경향을 가진 하나의 명백한 실례로 Web열람기들을 들수 있다. 어떤 기계에서 동작하고 있는 쏘프트웨어는 그 기계에 보관되여 있지 않아도 된다. 그것은 가까운 국부봉사기로부터 올수도 있고 또는 인터네트상의 어디엔가 있는 Web봉사기로부터 올수도 있다. 그러므로 《콤퓨터보안》과 《IT보안》이라는 말을 큰 혼란없이 같은 뜻으로 사용할수 있다.

얼핏 보면 《보안》이라는 말은 명백한 개념인것처럼 느껴 진다. 그러나 보안은 그에 꼭 맞는 의미를 찾으려 할수록 점점 더 애매해 지는 그러한 개념들중의 하나이다. 이미 콤퓨터보안에 대한 정의를 만들고 또한 그 정의들을 정정하는데 많은 노력이 돌려 졌다. 그러나 이러한 정의들은 거의 모두 포괄범위가 너무 협소하다든가 혹은 콤퓨터보안의 범위를 벗어 나 콤퓨터과학의 령역에 침입하고 있다는 비난을 받고 있다.

1. 보안

보안이란 대체로 재산의 보호를 말한다. 이 정의는 자기의 재산과 그것의 가치를 알고 있어야 한다는것을 암시한다. 이러한 일반적인 고찰은 물론 콤퓨터보안과 포괄적인 정보보안전략의 한부분인 위험분석(risk analysis)에서도 역시 옳은것으로 된다. 그러나 이러한 론제는 특별히 보호적인 대책에 초점을 둔 이 책의 범위를 벗어 나는것으로 된다.

보호대책들을 크게 갈라 보면 다음과 같이 구분된다.

- **예방** (Prevention):재산이 손상되는것을 막기 위한 대책을 취한다.
- **검출** (Detection):재산이 손상되였을 때 언제 어떻게 손상되였는가 그리고 누가 손상을 입혔는가를 검출할수 있도록 대책을 취한다.
- **반작용** (Reaction):재산을 회복시키거나 또는 재산을 손상으로부터 재생시키도록 하는 대책을 취한다.

이것을 례증하기 위하여 자기 집에 보관한 가치 있는 물건들을 보호하는것을 생각하자.

- **예방:** 도적이 집에 들어 가기 힘들게 문에 쇠를 채우고 창문빗장을 지르며 재산둘레에 벽을 쌓고 중세기적성새와 같이 물홈을 파는 등 보호대책을 취한다.
- **검출:** 어떤 물건을 도적 맞히였을 때 거기에 그것이 없으면 도난 당했다는것을 알것이고 침입이 일어 날 때 도난경보가 울리며 유선텔레비죤은 침입자를 식별해 내도록 하는 정보를 제공해 줄수 있다.
- **반작용:** 경찰을 부를수도 있고 도난 당한 물건을 다른것으로 교체하기로 결심할수 있으며 또는 경찰이 도난 당한 물건을 되찾아 돌려 줄수도 있다.

실세계에서의 이와 같은 실례들이 콤퓨터보안의 원리들을 설명하는데 도움을 줄수는 있지만 물리적인 보안과 콤퓨터보안을 동일시하는것이 언제나 타당한것은 아니다. 일부 용어들은 정보기술들에 리용되면 완전히 다른 뜻으로 된다. 콤퓨터보안의 령역에 보다 알맞는 실례를 보기 위해 인터네트상에서 주문을 할 때 신용카드번호의 리용을 고찰해 보자. 어떤 협잡군이 다른 사람의 신용카드번호를 리용하여 상품구입을 시도할수 있다. 이때 자신을 보호하기 위해서 어떻게 할수 있겠는가?

- **예방 :** 주문을 할 때 암호를 리용한다. 신용카드주문을 접수하기전에 방문객에 대한 일련의 검사를 진행하도록 상점에 요구하며 자기의 카드번호를 인터네트에서 리용하지 않는다.
- **검출:** 자기가 인정하지 않은 상품거래가 자신의 신용카드상태에 나타난다.
- **반작용:** 새로운 카드번호를 요구할수 있고 협잡상품거래의 값은 카드보관자나 또는 협잡군이 상품구입을 한 상점 또는 카드발행자가 보상한다.

이 실례에서 협잡군이 카드번호는 《도적질》하였지만 카드자체는 아직 주인에게 있다. 이것은 카드자체를 도난 당한 경우와는 다르다. 영국과 같은 법체제하에서는 협잡군이 신용카드번호를 몰래 알아 낸것에 대해서는 법적책임을 씌울수 없게 되여 있다. 결국 새로운 위협에 대처하기 위해 새로운 법이 채택되여야 한다.

이러한 조사의 계속으로 비밀정보들을 보호하기 위한 방안들을 생각해 보자. 보통 비밀은 그것이 로출되는 때에야 루설되였다는것을 알게 될것이다. 일부 경우에는 피해가

돌이킬수 없는것일수도 있다. 즉 경쟁자가 다른 사람이 여러해 품들여 개발한 제품설계를 손에 넣고 그보다 먼저 제품을 시장에 내놓아 리득을 볼수 있다. 이와 같은 경우에는 예방이 유일한 수단이다. 이것은 또한 어째서 력사적으로 콤퓨터보안이 비밀정보의 로출을 막는데 많은 주의를 돌려 왔는가를 설명해 준다.

2. 콤퓨터보안

콤퓨터보안에 대한 견해를 세우기 위해 우선 정보라는것이 어떻게 손상될수 있는가를 고찰해 보자. 대부분의 정의는 다음과 같은 3가지 측면을 포함한다.

- **기밀성**(Confidentiality): 권한 없는 자에 의한 정보의 로출을 예방.
- **완정성**(Integrity): 권한 없는 자에 의한 정보의 변경을 예방.
- **리용성**(Availability): 권한 없는 자에 의한 정보나 자원접근에 대한 보류의 예방.

이제는 곧 이러한 론제들의 우선권에 대한 론의를 시작할수 있으며 이 항목들을 다시 순서화할수 있다. 그러나 이것은 결코 완전한것이 아니며 만일 통신을 념두에 둔다면 확실성(authenticity), 전자상업과 같은 응용프로그람에 관심이 있다면 책임추적가능성(accountability)과 같은 점들을 추가할수 있다.

이러한 일반적인 준위에서조차 일련의 보안측면들에 대한 엄밀한 정의에서의 의견상이를 볼수 있다. 그러므로 이 책에서는 흔히 어떤 정의에 대한 참고문헌을 주어 거기에서 문맥이 명백해 지도록 한다. 보안평가기준(제9장)에 대해서는 US Trusted Computer System Evaluation Criteria(OrangeBook[112]), European Information Technology Security Evalution Criteria(ITSEC[117]), Canadian Trusted Computer Product Evaluation Criteria(CTCPEC[150])와 같은 책들을 참고하기 바란다. 실례로 우의 정의들은 ＩＴＳＥＣ로부터 인용한것이다.

3. 기밀성

력사적으로 보안과 비밀은 밀접히 련관되여 있다. 현재도 많은 사람들은 콤퓨터보안의 기본목적이 중요한 정보를 권한 없는 사용자들이 읽어 내지 못하게 막는것이라고 생각하고 있다. 일반적으로 권한 없는 사용자들은 중요한 정보(민감정보)를 알지 못하게 해야 한다. 기밀성(privacy, secrecy)은 콤퓨터보안의 이러한 측면을 말한다. 사적비밀(privacy)과 공적비밀(secrecy)이라는 용어들은 때때로 개인적자료의 보호(privacy)와 어떤 조직에 속하는 자료의 보호(secrecy)를 구분하기 위하여 사용된다. 기밀성은 잘 정의된 개념이며 특히 그것이 물리적보안에서는 맞다들리지 않았던 새로운 문제들을 일으키므로 콤퓨터보안에서는 자주 이 화제에 집중하여 연구를 진행한다. 때때로 보안과 기밀성은 동의어로 리용되기도 한다.

종이문서인 경우에는 어떤 문서에로의 접근을 그 문서를 읽도록 허락된 사람들의 목록을 만들어 놓으면 쉽게 통제할수 있다. 얼마간 이상하게 생각하겠지만 콤퓨터보안에서는 기밀성을 시행할 때 쓰기조작도 통제해야 한다. 여기에 대해서는 제4장 2절에서 더 심화된 내용을 볼수 있다.

4. 완정성

완정성에 대한 간결한 정의를 주는것은 매우 어려운 일이다. 일반적으로 완정성은 있어야 하리라고 생각하는것은 다 있다는것을 확신할 때 쓰인다. 이러한 정의는 사실을 나타내지만 별로 도움이 되지 못한다. 콤퓨터보안의 한계내에서는 완정성을 권한이 없는 쓰기의 방지를 다루는것으로 정의할수 있다. 이러한 해석에서 완정성과 기밀성은 쌍대적인 개념이며 여기서는 이 두가지 목표를 달성하는데 류사한 기술들을 리용할것으로 생각할수 있다.

그러나 더 나아가서 《어떤 일을 하도록 권한을 부여하는것》과 《정확한 절차를 따르는것》과 같은 문제들도 역시 완정성이라는 용어안에 포함된다. 이 정의는 완정성을 속성으로 선언한 클라크와 윌슨의 론문[32]에서 인용한것이다.

> **완정성**(Integrity): 회사의 재산이나 회계기록이 잃어 지거나 손상되는 방법으로는 비록 권한이 있다고 해도 체계의 어떤 사용자도 자료항목을 수정하는것을 허락할수 없다.

완정성을 권한이 없는 모든 행위를 막는것이라고 한다면 기밀성은 완정성의 한개 부분으로 된다. 지금까지 통제해야 할 사용자의 행위를 서술하는것에 의해 보안을 정의하였다. 체계적인 관점으로부터 완정성을 정의할 때 체계의 상태에 집중하면 더 좋은 결과를 얻게 된다. 자료완정성에 대한 《오렌지부크》(Orange Book)의 정의는 정확히 다음과 같다.

> **자료완정성**: 콤퓨터화된 자료가 원천문서의 자료와 같고 우연적이거나 혹은 고의적인 변경이나 파괴에 로출되지 않을 때 존재하는 상태.

여기서 완정성은 외면적인 일관성과 같은 말이다. 콤퓨터체계에 보관된 자료는 콤퓨터체계밖에서의 일련의 현실성을 정확히 반영하여야 한다. 물론 어떤 콤퓨터체계에 보관된 자료가 정확히 현실성을 반영하는것은 매우 중요하지만 그 콤퓨터체계의 내적인 기구들에 의해 이 속성을 담보하는것은 거의 불가능하다.

더우기 정보보호의 다른 령역들은 완정성에 대한 자기나름의 견해를 가진다. 실례로 통신보안에서 완정성은 《의식적인 속임수와 우연적인 전송오유를 포함하는 전송된 자료의 변경, 삽입, 삭제, 반복에 대한 검출과 정정》을 가리킨다.

누구에게도 변경할 권한을 주지 않았을 때 권한이 없는 변경의 특별한 경우로서 의식적인 변경을 고찰할수 있다. 그러나 권한을 부여하는 구조가 있는가 없는가 하는것이 해결해야 할 문제의 본성과 여러가지 보호기구들에 영향을 미치기때문에 그에 대한 론의는 략하기로 한다.

완정성은 흔히 다른 보호속성들을 위해서 필수적인것이다. 실례로 공격자는 조작체계나 또는 조작체계가 참조하는 접근조종표를 수정하여 기밀성조종을 못하게 하려고 시도할수 있다. 따라서 기밀성을 보장하기 위하여 조작체계의 완정성 또는 접근조종표의 완정성을 보호해야 한다.

마지막으로 보안과 리용성을 완정성의 부분으로 취급하는 보다 일반적인 완정성의 정의들도 있다는것을 지적하여 둔다.

5. 리용성

여기서는 CTCPEC에서 주어 진 정의를 인용하기로 한다.

리용성(Availability): 제품의 봉사가 필요할 때 부당한 지연이 없이 접근할수 있는 속성.

ISO/OSI의 통신보안을 위한 보안구조인 국제규격 ISO7408-2[51]도 거의 동일한 정의를 주고 있다.

리용성: 권한을 가진 실체에 의한 요구에 따라 접근할수 있고 리용할수 있는 속성.

리용성은 콤퓨터보안의 전통적인 한계를 넘어서 다른 부분들과 대단히 많은 관계를 가지고 있다. 실례로 리용성을 개선하기 위해 리용된 공학기술들은 장애허용계산과 같은 다른 령역들로부터 취해 진다. 보안에서는 나쁜 마음을 가진 공격자가 정당한 사용자들이 자기의 체계에 정당하게 접근하려는것을 막을수 없다는것을 담보하여야 한다. 즉 봉사거절을 방지하여야 한다. 이를 위해 ISO-7498-2의 정의를 참고한다.

봉사거절: 자원에로의 권한이 부여된 접근에 대한 차단 또는 시간림계조작의 지연.

최근에 인터네트에서는 공격자가 접속요구들로 봉사기를 압박하여 그것을 손 쉽게 무능하게 만드는 밀물공격이라는 사건이 있었다. 많은 경우에 리용성은 콤퓨터보안의 가장 중요한 측면이지만 이 문제를 취급하기 위한 보안기구들은 아직 제안되지 않았다. 사실상 지나치게 제한적인 보안기구들은 사용자들자신도 봉사거절 당하게 할수 있다.

6. 책임추적가능성

앞에서 콤퓨터보안의 전통적인 3가지 령역을 보았다. 돌이켜 보면 그것들은 모두 접근조종의 서로 다른 측면들을 취급하는것이고 바라지 않는 사건들을 미리 막는데 중점을 두고 있다는것을 알수 있다. 그러나 정당하지 않은 행위들을 다 막는다는것은 거의 불가능하다. 우선 권한을 가진 행위들이 보안위반으로 될수 있다. 다음으로 보안체계에는 공격자들이 침입할 길을 찾을수 있게 하는 약점이 있다. 그러므로 또 새로운 보안요구가 나서게 된다. 사용자들은 자기의 행위에 대하여 책임져야 한다. 이 요구는 전자상업거래와 같은 분야에서 특별히 중요하며 오렌지부크와 같은 문건들에 이미 정의되여 있다.

책임추적가능성(Accountability): 보안을 침해하는 행위들에 대해 책임 있는 부분까지 추적할수 있도록 검열정보를 선택적으로 유지하고 보호해야 한다.

그러기 위하여 체계는 사용자를 식별하고 인증하여야 한다. 또한 보안에 관계되는 사건들의 검열궤적을 유지하여야 한다. 만일 보안침해가 일어 나면 검열궤적으로부터의 정보는 범죄자와 그가 체계를 손상시키기 위하여 취한 걸음들을 식별해 낼수 있게 한다.

7. 믿음성과 안전성

보안에 대한 론의에서는 우발적인 고장에 관계되는 믿음성과 체계고장이 환경에 주는 영향에 관계되는 안전성 (이것들은 체계가 불리한 조건에서도 제대로 동작하여야 하는 정황을 취급한다.) 과 같은 콤퓨터분야의 다른 령역들도 언급한다. 거기에는 몇가지 리유가 있다. 첫째 리유는 개념에서의 겹침이다. 보안이 믿음성의 한 측면이기도 하며 혹은 반대로 믿음성이 보안의 한 측면으로도 된다. IFIP WG10.4는 단일화한 개념으로 의존성(dependability)을 받아 들이고 보안, 믿음성, 완정성, 리용성을 의존성의 측면들로 취급함으로써 이 난문제를 해결해 보려고 시도하였다[86].

의존성(Dependability): 콤퓨터체계가 제공하는 봉사에 대해 정당하게 믿음이 가는 그 콤퓨터체계의 속성. 여기서 체계가 제공하는 봉사라는것은 사용자가 느끼는 그 체계의 거동이다. 사용자는 그 체계와 호상작용하는 또 하나의 체계(실체, 인간)이다.

둘째 리유는 동시에 하나이상의 문제점들을 취급해야 하는 응용들이 있기때문이다. 실례로 안전성림계응용에서의 콤퓨터체계를 생각해 보자. 때때로 이 체계의 사용자들은 비상사태에 대처해야 한다. 보안조종은 고의적으로 재난을 일으키는 침입자들을 막아야한다. 침입검출체계는 낯선 거동모양을 보고 공격을 식별하려고 시도한다. 그런데 비상사태에 대한 반작용도 역시 낯선것으로 나타날수 있으며(비상사태는 매우 드문 사건이다.) 그래서 심각한 (림계)상황에서는 침입검출체계가 정당한 작용을 공격으로 잘못 인식하고 비상사태처리기구의 작용을 방해하는 보안기구를 시동시킴으로써 문제를 혼란시킬수 있다. 일반적으로 보안문제는 보호하려는 응용의 다른 요구들과 독립적으로 취급하지 말아야 한다.

결국 두 령역에서 다같이 류사한 공학적수법들이 리용된다. 실례로 보안쏘프트웨어를 평가하는 기준과 안전림계쏘프트웨어를 평가하는 기준은 많은 류사성을 가진다. 일부 전문가들은 최종적으로 하나의 기준만이 있어야 한다고 보고 있다.

8. 콤퓨터보안에 대한 정의

이 책에서는 콤퓨터보안에 대한 다음의 정의를 채용한다.

콤퓨터보안(Computer Security): 콤퓨터보안은 콤퓨터체계사용자들의 권한 없는 행위들에 대한 방지와 검출을 취급한다.

이 정의에서는 적절한 권한할당과 접근조종의 개념이 본질적이다. 적절한 권한할당은 보안방책 례하면 어떤 행위들이 허락되며 어떤 행위들이 금지된다는것을 나타내는 규칙들의 모임(보안방책)이 있다는것을 전제로 한다. 보안방책에서의 령역(domain)은 그 방책에 의해서 지배되는 어떤 실체들의 모임 례하면 사용자들, 자료객체들, 기계들 등의 모임이다. 보안의 정의에 부당한 행위의 영향을 바로 잡는것도 포함시킬수 있지만 이 측면은 중요하지 않다.

전문용어에 대한 이상의 개념적인 론의들로부터 얻은 기본결론은

1. 보안의 정의는 하나뿐이 아니다.
2. 문서를 읽을 때 문서에 언급된것과 보안에 대한 자기의 견해를 혼돈하지 않도록 주의해야 한다.
3. 보안에 대해 명백한 해석을 정의하려고 시도하는데 많은 시간이 소비되고 있다.

제2절. 콤퓨터보안의 기본난점

콤퓨터보안에 의존하는 사용자들의 수가 초기에 기밀자료를 취급하는 몇몇 조직들이였던것이 지금은 인터네트에 련결된 모든 사람들로 늘어 난것만큼 콤퓨터보안에 대한 요구는 완전히 변화되였다. 이 변화는 다음과 같은 기본난점을 낳게 하였다.

즉 보안을 모르는 사용자들은 자기나름의 보안요구를 가지고는 있지만 보통 보안에 대해 잘 모른다.

이 난점(dilemma)은 보안평가를 위한 현재 전략들에서 명백히 나타난다. 보안평가에서는 제품이 약속된 보안봉사를 제공하는가를 확인해야 한다. 그러자면 보안체계의 기능이 규정되여야 하며 보안조종이 효과적이며 침해시도를 견디여 낼것이라는것을 보증해야 한다.

오렌지부크는 보안제품(조작체계)을 평가하기 위한 첫번째 안내서로서 콤퓨터보안의 발전에 큰 영향을 주었다. 여기서는 기능과 보증이 미리 정의된 클라스들로 한데 묶어져 있으며 사용자들은 다만 이 묶음안내로부터 선택만 할수 있다. 아직까지 많은 제작자들은 자기 제품의 보안준위를 오렌지부크에서의 해당한 등급을 주는것으로 지적하고 있다. 그러나 오렌지부크는 신축성이 없으며 콤퓨터망이나 자료기지관리체계의 보안을 평가하는데 그리 좋은것은 아니다.

이로부터 보다 유연한 기준모임에 대한 요구가 제기되였으며 ITSEC에서 이 요구에 대답을 주었다. 여기서는 기능과 보증이 매우 특수한 평가목적(TOE:targets of evaluation)도 표현할수 있도록 분리하였다. 보안을 모르는 사용자는 특별한 TOE의 감각을 가지고 서로 다른 TOE들에 의하여 평가된 제품들을 비교하면 된다.

콤퓨터보안의 기본난점은 많은 형태로 나타난다. 이것을 해결하는것이 콤퓨터보안에서 현재 가장 절박한 문제이며 그것은 쉬운 일이 아니다.

이 기본난점에 비하면 보안과 사용자편리성사이의 모순관계는 쉽게 해결되는 공학적인 이률배반의 문제이다. 성능에 대한 보안의 영향은 다양하다.

- 보안기구는 보충적인 콤퓨터자원들을 요구한다. 그 비용은 쉽게 계산할수 있다.

- 보안은 사용자들의 습관된 작업방식들과 대립된다. 쓰기 불편하거나 적당치 않은 보안규칙들은 생산성을 감퇴시킨다.
- 노력은 보안을 관리하는데 바쳐져야 한다. 그러므로 보안체계의 구매자들은 흔히 가장 좋은 도형사용자대면부를 가진 제품을 선택한다.

보안에는 응당 적절한 비용이 지출되여야 한다. 보안을 고려하지 않은 비용의 평가는 위험분석의 한계내에서이다. 위험분석은 보안관리의 중요한 측면이지만 이 책의 범위밖의것이다.

제3절. 자료와 정보

콤퓨터보안은 정보와 자원에 대한 접근을 조종하는것이다. 그러나 정보에 대한 접근을 조종하는것은 때때로 매우 어려워 질 때가 있으므로 흔히 보다 수월한 자료에 대한 접근조종으로 바뀌운다. 자료와 정보사이의 구별은 미묘한것이지만 그것이 보안에서 일련의 보다 어려운 문제들의 근원으로 된다. 자료는 정보를 표현한다. 정보는 자료의 (주관적인)해석이다.

자료: 우리의 개념과 실세계의 일정한 측면을 묘사하기 위한 약속에 따라 선택된 물리적현상을 자료라고 한다. 자료에 부여한 뜻(의미)을 정보라고 한다. 자료는 정보를 전송하고 축적하기 위해서와 형식적인 규칙들에 따라 처리하여 새로운 정보를 유도해내기 위하여 사용된다.

정보와 대응하는 자료사이에 밀접한 련관이 있을 때 이 두가지의 취급방법은 매우 류사한 결과를 준다. 하지만 언제나 이렇게 되는것은 아니다. 실례로 잠복통로를 통하여 정보를 송신할수 있다(제4장 2절). 이때 자료는 접근요구에 대한《yes》나 《no》응답이고 수신된 정보는 중요한 파일의 내용이다. 다른 하나의 실례는 통계자료기지에서의 추론문제이다(제14장 4절).

이 문제를 간단히 보기 위해 납세신고서(Inland Revenue)자료기지를 고찰하자. 이 자료기지는 개인기록들에 접근할수 있는 세금검열관들에 의해 리용될뿐아니라 일반적인 계획작성의 목적으로 국가공무원들에 의해서도 리용된다. 그러므로 그들은 납세신고서들의 통계적개요들에 접근하여야 하지만 개인적기록들을 읽을 필요는 없다.

개인적인 기록들을 보호하기 위해 자료기지관리체계가 충분히 큰 자료모임에 대해서만 통계적질문을 허락한다고 가정하자. 그러면 하나의 기록만 차이나는 두개의 큰 자료모임에 대한 질문으로부터 결과들을 결합하는것이 가능할것이다. 이리하여 그 자료에 직접 접근함이 없이도 하나의 개인적기록에 대한 정보를 추출할수 있다.

제4절. 콤퓨터보안의 원리

지금까지 우리는 콤퓨터보안이 《로케트과학》과 같은 대단히 복잡한 문제라는것을 보았다. 그러나 놀랄 필요는 없다. 만일 조직적인 방법으로 콤퓨터체계의 보안특징들을 실현할 기회가 주어 진다면 쏘프트웨어(체계)개발에 대한 숙련된 방법들과 몇가지 본질적인 보안원리들에 대한 상세한 리해가 큰 도움을 줄것이다. 그러나 보안관련의 아무런 고려도 없이 채택된 설계결심에 의해 이미 복잡하게 된 체계에 때늦게라도 보안을 추가하여야 한다면 문제는 힘들어 질것이다. 유감스럽게도 대체로는 후자의 경우가 많다.

이제 콤퓨터보안의 몇가지 근본적인 설계정수들을 제기하겠다. 이 설계결심들은 이 책에 있는 표현들을 구조화하는 골격을 제공한다. 그림 1-1은 콤퓨터보안을 위한 설계공간의 기본차원을 보여 주고 있다. 수평축은 보안방책(제1장 4절 1)이 어디에 초점을 두어야 하는가를 나타내며 수직축은 보호기구가 실현되는(제1장 4절 2) 콤퓨터체계의 층을 나타낸다.
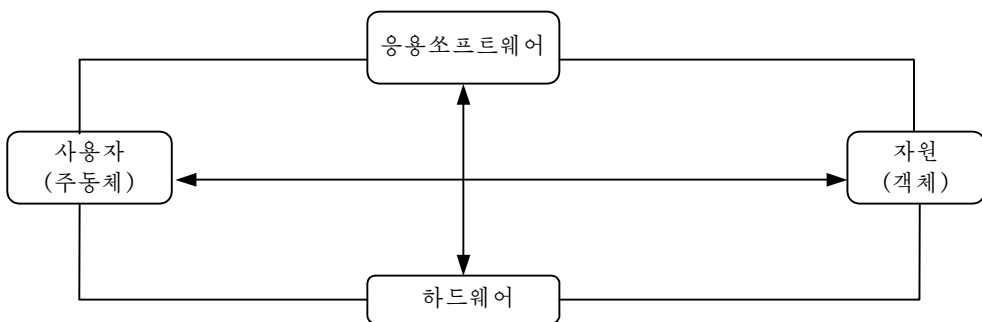
그림 1-1. 콤퓨터보안의 차원

1. 조종의 초점

제1장 1절 4에서 준 완정성의 정의를 다시 보자. 완정성은 주어 진 규칙묶음에 따라야 한다. 규칙들로는 다음과 같은것들이 있을수 있다.

- 자료항목들의 형식과 내용에 대한 규칙(내부일관성): 실례로 어떤 규칙이 구좌（account）자료기지에서 잔고(balance)마당들은 하나의 옹근수를 포함하여야 한다고 규정될수 있는데 이와 같은 규칙들은 그 자료기지항목에 접근하는 사용자나 그 자료기지항목우에서 수행되는 조작들에 의존하지 않는다.
- 자료항목우에서 수행될 조작들에 대한 규칙: 실례로 어떤 규칙이 《구좌열기》,《잔고검사》, 《꺼내기》, 《저금》조작들만이 구좌자료기지에 있는 잔고마당들에 대한 접근을 가지며 은행사무원들만이 《구좌열기》를 실행할수 있다고 규정될수 있는데 이와 같은 규칙들은 사용자와 자료항목에 의존한다.
- 어떤 자료항목에 대한 접근이 허락된 사용자에 대한 규칙: 실례로 어떤 규칙을 구좌소유자와 은행사무원들만 구좌자료기지에 있는 잔고마당들에 접근할수 있다는것으로 규정할수 있다.

여기서 하나의 중요한 일반적관측결과를 얻었으며 첫번째 설계원리에 도달하였다.

첫번째 설계결심: 주어 진 응용에서 콤퓨터체계에 있는 보호기구들이 무엇에 초점을 두는가? 자료인가? 조작인가? 사용자인가?

보안조종을 적용할 때 이러한 선택들이 초보적인 설계결심인것이다. 조작체계들은 전통적으로 자료(자원들)를 보호하는데 초점을 모아 왔다. 현대응용들에서는 사용자의 행위를 통제하는것이 보다 더 자주 제기된다.

2. 사람-기계척도

그림 1-2는 콤퓨터체계의 간단한 계층모형을 보여 준다. 이 모형은 다만 일반적인 안내를 위한것이다. 그러므로 이 모형이 모든 콤퓨터체계들에 있는 층들을 다 표현하는 것은 아니며 이 모형에 있는 5개 층보다 더 많은 층으로 구분할수 있는 체계도 있다.

응용
봉사
조작체계
OS 핵심부
하드웨어

그림 1-2. IT체계의 계층들

- 사용자는 전문적인 응용요구에 꼭 맞게 만들어 진 응용프로그람들을 실행시킨다.
- 응용프로그람들은 자료기지관리체계(DBMS)나 객체참조중개기(object reference broker-ORB)와 같은 일반용쏘프트웨어제품에 의해 제공된 봉사를 리용할수 있다.
- 이러한 쏘프트웨어제품들은 파일 및 기억관리를 수행하며 인쇄기나 I/O장치들과 같은 자원들에 대한 접근을 조종하는 OS(조작체계)상에서 동작한다.
- 조작체계는 처리기와 기억기에로의 모든 접근을 조정하는 핵심부를 가질수 있다.
- 하드웨어 즉 처리기와 기억기는 그 콤퓨터체계안에 있는 자료를 물리적으로 기억하고 처리한다.

보안조종은 이러한 층들중 임의의 층에 자리 잡을수 있다. 지금까지 두번째 근본보안원리의 차원들을 설명하였다.

두번째 설계결심: 콤퓨터체계의 어느 층에 보안기구를 배치하겠는가?

현재 보급되고 있는 보안제품들을 조사하면 하드웨어층으로부터 응용쏘프트웨어층에 이르기까지 이 모형의 매층에서의 보안기구들을 볼수 있다. 설계자의 과제는 매개 보안기구에 맞는 층을 찾아 내는것과 또 매개 층에 맞는 보안기구를 찾아 내는것이다.

이제 그림 1-3과 같이 제일 바깥층은 응용기구, 중심은 하드웨어기구로 동심보호고리를 이룬 콤퓨터체계의 보안기구를 놓고 새로운 보안원리를 다시한번 보기로 하자. 중심쪽에 있는 기구일수록 보다 일반적이고 보다 콤퓨터지향적이며 자료에 대한 접근조종과 관련되여 있을 경향성이 보다 크다. 바깥쪽의 기구일수록 보다 개별적사용자요구들을 처리하는데 적합하다. 우리의 첫 두개의 보안원리들을 조합하면 그림 1-4와 같은 사람-기계척도를 얻을수 있다.
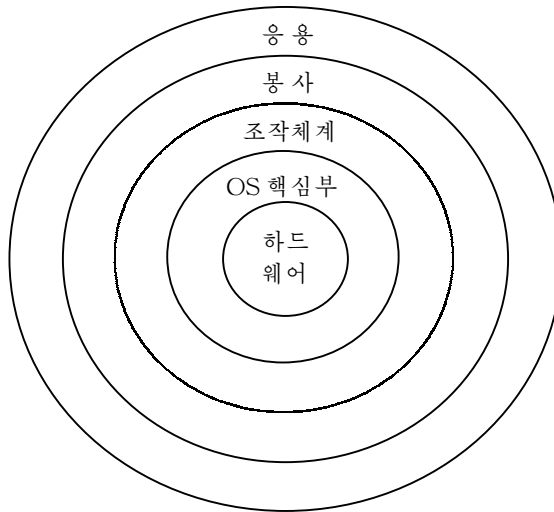
그림 1-3. 보호기구들의 동심원모형

그림 1-4. 접근조종기구들을 위한 사람-기계척도

3. 복잡성과 담보

사람-기계척도상에서 보안기구의 위치는 흔히 그의 복잡성과 밀접히 관련된다. 중심에서는 단순하고 일반적인 기구들을 볼수 있지만 응용층에는 특징이 풍부한 보안기능들이 시끄러울 정도로 많다. 이로부터 또 하나의 결심이 필요된다.

세번째 설계결심: 특징이 풍부한 보안환경에 비하여 단순한것(그리고 보다 높은 준위 담보)을 더 좋아 하는가?

이 결심은 콤퓨터보안의 기본난점과 관련된다. 간단하고 일반적인 기구는 특수한 보호요구들에 잘 맞지 않으며 반대로 풍부한 보안환경을 리용하려면 사용자들이 보안전문가로 되여야 한다. 보안을 모르는 사용자들은 승산이 없는 정황에 놓이게 된다.

높은 준위 담보을 얻기 위해서 보안체계는 면밀하게 그리고 될수록 철저히 시험되여야 한다. 이로부터 복잡성과 담보사이에는 명백한 이률배반의 관계가 있다. 보다 높은 준위 담보를 얻으려면 체계는 보다 간단해야 한다. 결과적으로 다음과 같은 원리를 알수 있다.

풍부한 보안체계와 높은 준위 담보는 쉽게 맞아 떨어 지지 않는다.

높은 준위 담보는 체계적인 설계원리들에 기초하여야 한다. 실제에 있어서 콤퓨터보안은 자기의 가장 높은 담보준위를 얻기 위한 도구로서 형식적인 방법들을 일찌기 적용한 령역들중의 하나이다.

4. 집중 또는 분산조종

보안방책의 령역내에서는 일관한 조종을 실시하여야 한다. 만일 보안을 책임진 단일한 중심적인 실체가 있다면 일관성을 성취하기는 쉽지만 이 중심적인 실체가 성능을 제한하는 병목(bottleneck)으로 될수 있다. 반대로 분산된 경우는 보다 효과적일수 있지만 이때는 서로 다른 여러 요소들이 일관한 방책을 시행하도록 담보하는데 주의를 더 돌려야 한다.

네번째 설계결심: 보안을 정의하고 실행하는 과제들이 하나의 중심적인 실체에 주어지는가 아니면 한 체계에 있는 개별적인 요소들에 분산되는가?

분산체계보안에서는 이러한 질문들이 자연적으로 제기되며 두가지 선택방안의 실례들을 볼수 있다. 그러나 이 질문은 제4장 2절에서 취급하는 벨-라파둘라(Bell-Lapadula)모형의 위임(mandatory) 및 자유(discretionary)보안방책에서 보여 주는것처럼 대형콤퓨터체계의 상황에서도 역시 의미를 가진다.

제5절. 보안기구의 아래층

앞에서 우리는 담보에 대해서는 간단히 보고 주로는 가장 적절한 보안방책들을 표현하기 위한 선택안들을 보았다. 이제부터는 보호기구를 뚫고 들어 오려고 시도하는 공격자들에 대해서 보기로 한다. 매 보호기구는 보안경계(한계)를 정의한다. 보호기구를 손상시킴이 없이는 제대로 동작시킬수 없는 체계의 부분들은 이 경계밖에 놓인다. 보호기구를 무력하게 하는데 리용될수 있는 체계의 부분들은 이 경계안에 놓는다. 이러한 고찰은 제1장 4절 2에서 제안된 둘째 설계원리를 직접 확장할수 있게 한다.

다섯째 설계결심: 보호기구의 아래층에로 접근하는 공격자를 어떻게 막을수 있는가?

《아래층》에로 접근하는 공격자는 계속하여 더 우의 보호기구들을 파괴할수 있다. 실례로 조작체계의 체계특권을 얻게 되면 보통 봉사층이나 응용층에 있는 보안기구용의 조종자료를 포함하는 프로그람이나 파일들을 변경시킬수 있다. 물리적기억장치에로의 직접접근에 의해 조작체계의 론리적접근조종을 우회할수 있다. 아래에서는 이 점을 례증하기 위한 5가지의 실례들을 더 주게 된다. 보안기구들은 일정한 약점을 가지고 있으며 아래층으로부터 공격자들에게서 공격 받을수 있다. 그러나 콤퓨터보안기구를 적용할수 없는 단계에 이르렀을 때에도 물리적 또는 조직적인 보안대책을 세울수 있다(그림 1-5).
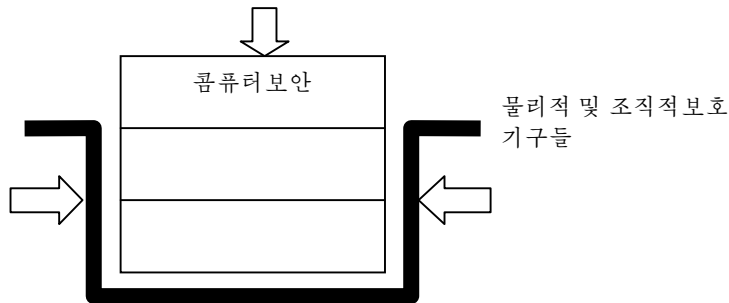
그림 1-5. 아래층에로의 접근은 물리적 및 조직적보안대책들을 통하여 조종된다

회복도구

만일 기억기의 론리적조직화가 일부 물리기억기장애에 인해 파괴되면 물리적표현은 아직 변하지 않았다고 해도 파일에 더는 접근할수 없다. Norton Utilities와 같은 회복도구들은 (물리적)기억기를 직접 읽고 파일구조를 회복하여 자료를 회복하도록 도울수 있다. 그러나 이와 같은 도구들은 론리적기억구조에 주의를 돌리지 않으므로 론리적접근조종을 회피하는데 리용될수 있다.

Unix 장치

Unix는 I/O 장치들과 물리적기억장치들을 파일들과 같이 취급한다. 따라서 이러한 장치들에 대해 파일들에 리용한것과 같은 접근조종기구들이 그대로 적용될수 있다. 만일 접근허락이 잘못 정의되면 실례로 읽기금지된 파일들을 포함하고 있는 디스크에로 읽기 접근이 주어 진다면 이때 공격자는 그 디스크내용을 읽고 그 파일들을 다시 구성할수 있다. Unix 보안에 대한 보다 상세한 정보는 6장에서 보게 된다.

객체재리용(기억기의 해방)

단일처리기다중프로그람체계에서는 몇개의 처리가 동시에 실행될수 있다. 그러나 어떤 한 순간에는 오직 하나의 처리만이 처리기를 《소유》할수 있다. 조작체계가 현재 실행중의 처리를 비능동으로 하고 다음번 처리를 능동으로 할 때마다 상태절환이

수행되여야 한다. 다음번에 실행이 계속되도록 하는데 필요한 모든 정보가 보관되여야 하며 새로운 처리와 관련되는 정보들이 설정되여야 한다. 보안의 리유로 하여 잔류기억 례하면 새 처리에 할당된 기억령역에 남아 있는 자료들을 무효화하여야 한다. 이것은 해방된 모든 기억위치들에 어떤 고정된 모양을 써넣는 조작에 의해서 또는 이미 그것이 씌여 진 위치들에만 새로운 처리의 읽기접근을 허락하는 방법으로 해결할수 있다.

여벌만들기

세심한 체계관리자는 정기적으로 여벌만들기를 진행한다. 그 여벌(Back up)테프에 손을 댈수 있는 사람이면 누구나 그 테프에 있는 모든 자료에 접근할수 있다. 이 경우에 론리적접근조종은 소용 없으며 여벌테프는 자료를 보호하기 위해 안전하게 자물쇠를 잠그어 보관하여야 한다.

기억쏟기

체계는 붕괴될 때 자기의 내부상태의 기억쏟기(core dump)를 창조하여 붕괴의 원인을 쉽게 식별할수 있게 한다. 만일 내부상태가 암호열쇠와 같은 민감한 정보들을 포함하고 있다면 그리고 기억쏟기가 누구나 읽을수 있는 파일에 기억되여 있다면 로련한 공격자는 보안을 쉽게 파괴할수 있을것이다.

이 장의 문헌안내

콤퓨터보안에 대해서는 많은 책들이 있다. 이 장의 론제에 대해 꼭 읽어야 할 문헌은 [130]이다. [145]는 콤퓨터보안을 위한 좋은 책이며 보안관리에 대하여 보다 깊이 서술하고 있다. [1]은 콤퓨터보안의 리론적요소들을 취급하고 있다. [56]에서는 1980년대에 보안조작체계를 구축하는데서 얻은 가써(Gasser,M)의 경험을 통하여 안전한 조작체계를 설계하는 기술들에 대한 완벽한 론의를 주고 있다. 콤퓨터보안에 대한 또 하나의 포괄적인 문헌은 [125]이며 여기서는 많은 가치 있는 참고서들을 지적하고 있다.

제 1장에 대한 참고서로는 [163]이며 접근경로에 대한 론의가 특별히 취급되였다. 이 책은 또한 재정부문들에서 일반적인 조작체계들의 보안특징들에 대해 흥미 있게 서술하였다. 특정한 조작체계들의 보안특징들에 대한 책들은 많지 않으며 또 그러한 체계를 관리하는 일부 사람들에게만 관계되는 문제점들을 취급하였으므로 보안을 실현하는 방법들에 대해서는 그리 좋은 책들이 못된다. 그러나 AS/400[122]에서는 조작체계에 대한 상세한 기술들을 주며 낮은 층에 있는 코드가 조작체계에 의해서 제공되는 보안을 어떻게 손상시킬수 있는가를 보여 준다. Unix 보안에 대한 좋은 책들은 [36,50,55,162]들이며 Windows NT에 대해서는 [63,139,149]를 참고할수 있다. 기타 전문적인 문제들에 대해서는 해당 장의 끝에 있는 참고문헌에서 찾아 보면 된다.

연구령역의 현 상태를 파악하기 위해서는 그의 력사도 알고 있어야 한다. 콤퓨터보안에서 많은 연구를 촉발시킨 두개의 중요한 론문들은 [156]과 [2]이다. 이 론문들을 리해하려면 콤퓨터보안의 초기력사를 요약한 [93]을 보는것이 좋다.

련습문제

1. 보안개념들을 정의하기 위한 연구에 대해서 말해 보시오. US TESEC프로그람, UK ITSEC계획, 카나다의 CTCPEC의 Web싸이트들을 참고하시오.

   http://www.radium.ncsc.mil/tpep/process/fag.html
   http://www.itsec.gov.uk
   ftp://ftp.cse-cst.gc.ca/pub/criteria/CTCPEC

   많은 주요 IT회사들도 역시 자기들의 Web싸이트들에 보안에 관한 폐지들을 가지고 있다.

2. 자료와 정보의 차이를 론하는 짧은 소론문을 쓰고 자료에 대한 접근을 조종하는것이 반드시 정보에 대한 접근을 조종하는것을 암시하는것은 아니라는것을 보여 주는 실례를 찾으시오.

3. 의학적기록들은 특별한 보안문제들을 요구한다. 어떤 사람의 의학적기록이 직결(on-line)로 접근될수 있다고 하자. 이 정보는 비밀에 속하며 파괴로부터 보호되여야 한다. 한편 비상시에는 그 사람을 치료하는 사람이면 누구든지 그의 기록에 접근할수 있게 하여야 한다. 그러자면 그는 자기의 기록을 안전하게 하기 위하여 보호, 검출, 회복을 어떻게 리용하겠는가?

4. 콤퓨터체계에 보관된 시험결과를 보호하기 위한 보안방책을 이끌어 내시오. 그 방책은 적어도 대학생들, 강의자들, 관리자들의 접근요구를 고려해야 한다.

5. 자기가 리용하는 콤퓨터체계에서 보안기구들을 잠재적으로 병합시킬수 있는 쏘프트웨어요소들을 구분하시오.

6. 토론: 훌륭한 도형사용자대면부가 보안제품을 구입하기 위한 적절한 척도인가?

7. 어떤 층에 있는 보안기구가 그 아래층에 접근한 공격자에 의해서 우회될수 있는 다른 실례들을 찾아 보시오.

8. 개인용콤퓨터(PC)보안을 해석할 때 적용할수 있는 보안경계들을 구분하시오. 이때 개인용콤퓨터가 있는 방, 개인용콤퓨터, 개인용콤퓨터안에 들어 있는 보안모듈이 보안경계안에 놓인다고 가정하시오.

제2장. 식별과 인증

보안체계는 봉사를 요구하는 사용자들의 신원을 추적하여야 한다. 인증이란 사용자의 신원을 확인하는 공정이다. 사용자를 인증하는데는 두가지 원인이 있다. 즉

- 사용자신원이 접근조종결심에 속하는 하나의 파라메터이다.
- 사용자신원은 검열자료일지에 보안관련사건들을 기입할 때 기록된다.

접근조종을 사용자신원에 기초하여 하는것이 항상 필요하거나 좋은것은 아니다. 검열기록에 있는 신원을 리용하는것이 훨씬 더 좋은 경우도 있다. 이 장에서는 식별과 인증을 현재의 콤퓨터체계들에서 규격화된것으로 본다. 분산체계들에서의 인증은 제10장에서 취급한다.

목적

- 이미 알려 진 기구들을 다시 보고 몇가지 일반적내용을 학습한다.
- 통과암호보호에 대한 기초를 얻는다.
- 보안기구들이 행정적대책에 의거할 때 효과적이라는것을 인식한다.
- 콤퓨터보안에서 추상화를 리용할 때의 위험성을 리해한다.

제1절. 사용자이름과 통과암호

보통 사용자는 콤퓨터에 등록가입할 때 처음으로 콤퓨터보안과 접촉하게 되며 사용자이름과 통과암호를 입력하라는 요청을 받는다. 첫번째 단계를 식별이라고 하며 사용자는 자기가 누구인가를 알린다. 두번째 단계를 인증이라고 하며 사용자는 자기가 그 사람이라는것을 증명한다. 단어《인증》(authentication)을 다르게 해석하지 않도록 명백히 다음과 같이 정의한다.

실체인증: 제기된 신원을 확인하는 공정.

일단 사용자이름과 통과암호를 입력하면 콤퓨터는 그것을 통과암호파일에 기억된 기입항목들과 비교한다. 유효한 사용자이름과 통과암호가 입력되였다면 가입을 실현한다. 만일 사용자이름이나 통과암호가 틀리게 입력되였다면 가입은 실패한다. 보통 이때 가입화면이 다시 나타나며 사용자는 재차 가입을 시도할수 있다.

일부 체계들은 실패한 가입시도회수를 기억하며 일정한 턱값에 도달하면 사용자등록자리에 자물쇠를 잠근다. 사용자가 사용하다가 놓아 둔 콤퓨터를 공격자가 리용할수 있는 기회를 줄이기 위해 작업의 시작뿐아니라 그 작업과정의 일정한 간격마다에서 인증을 요구할수 있다(반복인증).

사용자가 콤퓨터를 켜두고 지나치게 오랜 시간동안 놀고 있으면 자동적으로 화면을 잠그거나 또는 작업을 끝내도록 할수 있다.

반복인증은 콤퓨터보안에서 TOCTTOU(time of check to time of use- 검사시간 대 리용시간)로 알려 져 있는 문제를 취급한다.조작체계는 대화가 시작될 때 사용자신원을 검사할뿐아니라 좀 더 지나 대화기간사이에도 접근조종결심을 하기 위하여 신원을 리용한다.

이전에 사용자는 친절한 환영통보문과 접근하려는 체계에 대한 일련의 정보를 포함하는 화면에 사용자이름과 통과암호를 넣어야 하였다. 그러나 오늘 조심성 있는 체계관리자는 지내 많은 정보를 제시하지 않으며 환영통보문을 권한이 없는 사용자에 대한 경고로 교체하였다. 실례로 Windows NT는 법률적주의(legal notice)를 나타내는 선택을 제시한다. 사용자들은 가입하기전에 이 경고통보문에 응답하여야 한다.

오늘날 대부분의 콤퓨터체계는 첫번째 방어선으로서 사용자이름과 통과암호를 리용한 식별과 인증기구를 사용한다. 이 기구는 사용자들이 콤퓨터에서 작업을 시작하는 루틴의 완전한 한개 부분으로 되였다. 이와 같은 기구는 널리 리용되며 또 실현하기 그다지 어렵지 않다. 한편 유용한 통과암호를 얻는것은 콤퓨터체계에 대한 권한이 없는 접근을 하기 위한 공통적인 수법이다.

인증기구로서의 통과암호의 실제적보안을 검토해 보자. 다음과 같은 세가지 위협들이 존재한다.

- 통과암호의 추측
- 통과암호의 기만
- 통과암호파일의 손상

통과암호보호에서는 사용자가 중요한 역할을 한다는것을 잊지 말아야 한다. 자기의 통과암호를 친구에게 루설하거나 노트에 씀으로써 다른 사람에게 알려 질수 있다.

제2절. 통과암호의 선택

통과암호의 선택은 중요한 보안문제이다. 공격자가 우연히 유효한 통과암호를 추측하는것을 완전히 막을수는 없으므로 이러한 가능성을 될수록 낮추기 위해 노력해야 한다. 그러자면 공격자들이 기본적으로 다음과 같은 두가지 추측전략에 의거한다는것을 알아야 한다.

- **완전탐색**: 어떤 일정한 길이까지의 유효한 기호들의 가능한 조합을 모두 시도해 본다.
- **지능적탐색**: 제한된 이름공간을 통하여 탐색한다. 실례로 사용자이름, 친구들이나 일가친척들의 이름, 자동차상표, 자동차등록번호, 전화번호 등 사용자와 관계되는 통과암호나 또는 일반적으로 공통적인 통과암호를 알아 내려고 시도한다. 두번째 방법의 전형적실례는 직결사전으로부터 모든 통과암호를 알아 내려고 하는 사전공격(dictionary attack)이다.

그러면 어떻게 방어하겠는가?

- **통과암호의 설정**: 체계관리자 또는 사용자가 사용자등록자리(account)를 위한 통과암호를 설정하는것을 잊었다면 공격자는 통과암호를 추측하는 수고조차 당하지 않게 된다.
- **기정통과암호의 변경**: 체계들은 주어 질 때 흔히 《manager》(관리자)라는 기정통과암호와 《system》(체계)과 같은 기정등록자리를 가지고 있다. 이것은 현장기사의 체계설치를 돕는다. 그러나 그 통과암호를 변경시키지 않은채로 두면 공격자가 그 체계에로 쉽게 뚫고 들어 올수 있으므로 기정통과암호를 변경시켜야 한다. 이렇게 하였다해도 공격자는 특별히 특권을 준 등록자리에 대한 접근을 할수 있다.
- **통과암호길이**: 완전탐색을 좌절시키기 위하여 최소한의 통과암호길이를 미리 규정하여야 한다. Unix체계들은 최대통과암호길이를 가지며 8개 문자까지 설정할수 있다.
- **통과암호형식**: 통과암호에 대문자, 소문자기호들을 섞어 쓰며 자모가 아닌 기호들과 수자들을 포함한다.
- **명백한 통과암호를 피하기**: 공격자들은 널리 쓰이는 통과암호들의 목록을 가지고 있으며 사전공격은《명백한》 탐색범위를 충분히 확장하였다는것을 알아야 한다. 오늘날 거의 모든 언어들에 대한 직결사전(on-line dictionary)을 찾아 볼수 있다.

그러면 체계는 어떻게 통과암호보안의 개선을 도울수 있는가?

- **통과암호검사기**: 체계관리자는 일부 《약한》 통과암호사전들에 대하여 통과암호들을 검사하는 도구를 리용할수 있다. 이것은 체계에 대한 사전공격을 모방해서 미리 막을수 있게 한다.
- **통과암호생성기**: 일부 조작체계들은 우연적이지만 발음할수 있는 통과암호들을 만들어 내는 통과암호생성기를 포함하고 있다. 이때는 사용자들이 자기의 고유한 통과암호를 선택하는것은 허락되지 않으며 체계에 의해서 제안된 통과암호를 리용해야 한다.
- **통과암호의 로화**: 대부분의 체계들에서는 통과암호의 만기날자를 설정할수 있으며 규칙적인 간격으로 통과암호를 변경할것을 사용자에게 요구한다. 또한 사용자들이 이전의 통과암호를 다시 사용하는것을 막는 추가적인 기구들로서 례하면 이미 리용된 과거의 10개 통과암호목록 등이 있다. 물론 낡은 통과암호에 대해 충분한 회수의 변경을 함으로써 사용자들은 자기가 좋아 하는 통과암호를 다시 쓸수도 있다.
- **가입시도의 제한**: 체계는 실패한 가입시도회수를 감시하고 사용자등록자리를 완전히 잠그거나 또는 적어도 일정한 시간동안 가입시도를 막거나 정지시킬수 있다.
- **사용자에게 통지**: 사용자가 가입에 성공하면 체계는 이전의 마지막 가입시간과 그때부터 실패한 가입시도의 회수를 화면에 현시하여 사용자에게 최근에 시도된 공격에 대하여 통지할수 있다.

이와 같이 체계가 생성하는 대, 소문자들과 수자기호들이 혼합되여 있고 정기적으로 변화되는 긴 통과암호를 리용하였다고 하여 가장 높은 수준의 보안이 실현되는것은 아니다.

사용자들은 길고 복잡한 통과암호들을 기억할수 없다. 대신에 이러한 통과암호를 종이에 쓰게 되는데 그것을 통과암호를 노리는 침입자가 볼수 있다. 콤퓨터말단들에서 전송되는 표기들에서 통과암호를 찾아 내는것은 보안관리자들의 기본과제이다. 통과암호들

이 자주 변화될 때에도 역시 마찬가지이다. 이와 같이 통과암호관리도식의 엄격한 요구를 따르기는 어렵다. 이것을 아는 사용자들은 보다 쉽게 기억할수 있는 통과암호를 리용하려고 할수 있는데 그것은 오히려 공격자들에게 쉽게 추측될수 있다. 또한 자기가 좋아하는 통과암호에로 빨리 되돌아 오거나 또는 통과암호에 간단하고 의미 있는 변화를 가하는것으로 대치하려고 할수 있다. 만일 매달 통과암호를 바꾸어야 한다면 선택한 통과암호에 달(1월부터 12월까지의 두자리 수자 또는 JAN부터 DEC까지의 3개 문자중에서 선택한다.)을 추가하여 기억하기 쉬운 통과암호를 만들수 있다.이때 이 통과암호들중의 하나를 발견한 공격자는 다음것을 쉽게 예측할수 있다.

여기에 고려해야 할 또 하나의 측면이 있다. 사용자들이 모든 보안규정을 매우 엄격하게 지키며 쉽게 추측할수 있는 통과암호들은 피하고 통과암호를 노트에 기록하지 않았는데 후에 그것을 잊었다고 가정하자. 이것은 사용자의 사업을 혼란시킨다. 또 사용자는 새로운 통과암호를 얻기 위해 체계관리자와 접촉하여야 하는데 이것은 체계관리자의 사업을 혼란시키며 새로운 공격의 길을 열어 주는것으로 된다. 사용자와 체계관리자가 직접 만날수 없다면 전화를 통하여 새로운 통과암호를 합의할수 있다. 이때 체계관리자가 사용자에게 적절한 권한을 부여할수 있겠는가?

조작자를 협박하여 통과암호를 내놓게 하는것은 체계에로 침입하기 위하여 이미 시도되고 시험된 방법이다. 성공적인 공격들은 흔히 기술적묘술보다 사회적수법에 기초하고 있다.

보안기구들을 고립적으로 고찰하지 말아야 한다. 하나의 보안기구에 지나치게 많은 강조를 두면 사용자들이 자기 일을 하기 위해 보안을 피하므로 오히려 체계를 약화시킬수 있다. 통과암호와 함께 통과암호의 복잡성과 인간의 기억능력사이의 이률배반관계를 고찰하여야 한다.

제3절. 기만공격

사용자이름과 통과암호를 통한 식별과 인증은 일방적인 인증이다. 사용자는 통과암호를 대고 콤퓨터는 사용자의 신원을 확인한다. 그러나 사용자는 이 통과암호를 누가 받았는지 모른다. 선로의 다른쪽 끝부분 즉 상대방의 신원에 대하여 보증할수 없다.

이것은 실제의 문제로서 통과암호손상의 두번째 부류이다. 기만공격에서는 공격자가 일부 말단/워크스테이션에 가짜가입화면을 제시하는 프로그람을 실행시킨다. 이것을 의심하지 않는 사용자는 이 말단에 가입하려고 시도한다. 기만프로그람은 사용자를 정상가입차림표에서처럼 유인하여 사용자이름과 통과암호를 넣도록 요구한다. 입력된 사용자이름과 통과암호는 공격자에 의해 기억된다. 다음에 실행은 사용자에게로 넘어 가거나 가입이 (거짓)오유통보문으로 취소되며 기만프로그람은 끝난다. 조종은 조작체계에로 되돌아 가고 이때 사용자에게 진짜가입요구가 재촉된다. 사용자는 이 두번째 가입요구에 따라서 또다시 가입을 시도하며 통과암호가 루설되였다는것을 전혀 모르고 지나갈수 있다.

이와 같은 기만공격에 대처하여 어떻게 할수 있는가?

- 실패한 가입회수를 현시하여 사용자에게 이와 같은 공격이 있었다는것을 알려 줄수 있다. 만일 첫번째 가입에서 실패하였고 두번째 시도에서 가입하였는데 지난번 작업이후 실패한 가입시도가 없었다고 통지되면 의심스럽게 보아야 한다.

- 신용 받는 경로: 사용자가 기만프로그람이 아니라 조작체계와 통신하였다는것을 보증한다. 실례로 Windows NT는 조작체계가입화면을 불러 내는 안전안전주의조작렬 《CTRL+ALT+DEL》을 가진다. 사용자는 작업을 시작할 때 가입화면이 이미 나와 있다 해도 이 주의신호건을 눌러야 한다.
- 상호인증: 만일 사용자들이 자기들이 통신하고 있는 체계의 신원에 대한 엄격한 보증을 요구한다면(실례로 분산체계에서) 체계는 자신을 사용자에게 인증하여야 한다.

기만공격외에도 침입자는 통과암호를 찾아 내는 다른 수법들을 쓸수 있다. 여기서 한 등록가입의 서술은 매우 추상화되였다. 통과암호는 사용자로부터 직접 통과암호검사루틴에로 넘어 간다. 그러나 실제로 그것은 도중에 완충기나 완충기억기 지어는 Web페지와 같은 중간기억기들에 림시로 기억되게 된다. 이 기억위치들의 관리는 보통 사용자의 조종밖에 있으며 통과암호는 사용자가 예상하였던것보다 더 오래 머무를수 있다.

이 문제는 Web기초직결은행봉사(참고문헌[5])의 개발자들이 부닥쳤던 문제에서 잘 보여 준다. Web열람기들은 사용자가 최근에 방문한 페지들을 다시 볼수 있게 하는 정보를 보관한다. 사용자는 직결은행봉사를 리용하기 위하여 Web페지에 자기의 통과암호를 입력한다. 사용자는 자기의 사무를 마치고 은행응용프로그람은 닫았으나 열람기대화는 끝내지 않았다고 하자. 이때 말단에 있던 다른 사용자가 앞사람의 통과암호를 가지고 그 페지를 다시 볼수 있으며 그의 자격으로써 가입할수 있다.

이에 대한 예방책으로서는 은행거래후에 반드시 열람기를 닫는것이다. 이때는 사용자들이 기억관리활동에 관여할것이 요구된다. 이것은 객체재리용의 다른 하나의 실례이다(제1장 5절).

추상화는 쓸모 있는 동시에 위험하다. 추상화용어로 통과암호보안을 론의하는것은 편리하다. 통과암호가 IT체계에서 어떻게 처리는가를 알지 못하고 통과암호형식에 대한 방책이나 로화를 조사할수 있다. 그러나 이러한 추상화준위에서만 통과암호보안을 론의하는것은 위험하다. 실현약점은 최상의 보안방략을 손상시킬수 있다.

제4절. 통과암호파일의 보호

체계는 사용자의 신원을 확인하기 위하여 사용자가 입력한 통과암호를 통과암호파일에 기억된 값과 비교한다. 물론 이러한 통과암호파일은 공격자들의 목표이다. 통과암호파일의 내용을 암호화하지 않은채로 두거나 통과암호파일의 내용을 변경시키는것은 통과암호손상의 세번째 가능성을 초래한다. 지어는 암호화된 통과암호가 폭로되는데 대해서까지 근심할수 있다. 이때 사전공격은 비직결로 진행될수도 있는데 이때는 실패한 가입시도회수를 제한하는것과 같은 보호기구들은 쓸수 없게 된다. 통과암호파일을 보호하기 위해 다음과 같은 선택을 가진다.

- 암호학적인 보호.
- 조작체계에 의해서 시행되는 접근조종.

- 암호학적인 보호와 접근조종 그리고 사전공격을 감퇴시키기 위한 기타 개선된 방법들의 조합.

암호학적인 보호에서는 암호화알고리듬조차 필요 없다. 한방향함수(one-way function)가 그 일감을 수행하게 된다. 이제부터 다음의 정의를 리용한다.

한방향함수란 함수값을 계산하는것은 비교적 쉽지만 거꾸로 계산하는것은 매우 어려운 함수이다. 즉 x가 주어 지면 $f(x)$를 계산하기는 쉽지만 반대로 $f(x)$가 주어 질 때 x를 계산하는것은 어렵다.

제12장에서 한방향함수에 대해 상세히 취급한다. 한방향함수들은 기억된 통과암호를 보호하는데 리용되였다[158]. 통과암호 x대신에 함수값 $f(x)$가 통과암호파일에 기억된다. 사용자가 가입할 때 통과암호 x'를 입력하면 체계는 한방향함수 f를 적용하여 $f(x')$를 구하고 예측값 $f(x)$와 비교한다. 이 두 값이 일치하면 사용자는 성공적으로 가입하게 된다. 앞으로 우리는 통과암호에 한방향함수를 적용하고 있는 경우에 《암호화된》 통과암호에 속하는것으로 본다.

사전공격의 우려가 없다면 통과암호파일을 전체읽기가능으로 할수 있다. f가 적당한 한방향함수라면 $f(x)$로부터 통과암호 x를 재구성하는것은 불가능하다. 사전공격에서는 공격자가 사전에 있는 모든 단어들을 《암호화》하고 그 결과를 통과암호파일에 있는 암호화된 항목들과 비교한다. 일치하는것이 있으면 공격자는 사용자의 통과암호를 알게 된다. 한방향함수를 사전공격을 지연시키는데 리용할수 있다. Unix체계에서는 한방향함수 **crypt(3)**을 사용한다. 이것은 시작값으로는 모두 령인 블로크를, 열쇠로는 통과암호를 리용하여 약간 수정된 DES알고리듬을 25번 반복한다[104]. 물론 가입할 때 합법적인 사용자들에 대해서는 약간한 성능저하가 있으나 한방향함수를 속도에 대하여 최량화한다면 사전공격의 성능도 동시에 개선하는것으로 된다.

조작체계에 있는 접근조종기구들은 파일과 기타 자원들에 대한 접근을 적절한 특권을 가지는 사용자들로 제한한다. 즉 특권이 부여된 사용자들만이 통과암호파일에 대한 쓰기접근을 할수 있다. 통과암호가 암호학적방법에 의해 보호된것이라 해도 공격자는 그것을 변경시킴으로써 쉽게 다른 사용자들의 자료에 접근할수 있다. 특권이 있는 사용자들만으로 읽기접근을 제한하면 리론적으로는 통과암호들을 암호화하지 않고 기억할수 있다. 통과암호파일이 권한 없는 사용자에게도 요구되는 정보를 포함하고 있다면 통과암호를 암호화하여야 한다. 그러나 이러한 파일은 사전공격들에서 아직 리용될수 있다. 대표적인 실례로 Unix에서 **/etc/passwd**를 들수 있다. Unix의 최신판들은 공격으로는 공개적으로 접근할수 없는 파일에 암호화된 통과암호들을 기억한다. 이런 파일들을 그림자통과암호파일이라고 한다. 실례로 HP-UX는 그림자통과암호파일**/.secure/etc/passwd**를 리용한다.

읽기보호의 약한 형식은 전용(proprietary)기억형식이다. 실례로 Windows NT는 전용2진형식으로 암호화된 통과암호들을 기억한다. 공격자는 보안관련자료의 위치를 검출하는데 필요한 정보를 추측해 낼수 있다. 《애매성에 의한 보안》은 그자체로서는 그다지 강하지 않으나 그것을 통과암호의 암호화와 같은 다른 기구들에 추가할수 있다.

그러나 이와 같은 방어를 성공적으로 돌파하면 모든 부분을 파괴시킬수 있다는 위험이 있다. 1997년 초에 Windows NT의 통과암호보안이 파괴되였다는 소동이 일어 났다. 문제는 매우 심각하였다. 소동의 동기로 된것은 암호화된 통과암호들을 2진파일로부터 보다 읽기 쉬운 표현으로 변환하는 프로그람이 발표된것이였다. 소동의 후에 큰 일은 없었다.

만일 사전공격이 우려되는데 통과암호파일을 감출수 없다면 통과암호절임(salting)을 적용할수 있다. 통과암호를 기억시키기 위해 암호화할 때 추가적인 정보

(소금)를 암호화하기전의 통과암호에 첨가한다. 소금정보는 암호화된 통과암호와 함께 기억된다. 만일 두 사용자가 같은 통과암호를 가지고 있다고 해도 그것들은 암호화된 통과암호파일에서 서로 다르게 기입되게 된다. 결국 절임은 여러 사용자들의 통과암호를 일제히 탐색할수 없게 함으로써 사전공격을 지연시킨다.

이상에서 3가지 보안설계문제들을 보았다.

- 기구들의 조합은 보호를 개선할수 있다. 암호화와 접근조종은 통과암호파일들을 보호하는데 리용된다.
- 애매성에 의한 보안은 우연한 침입자만을 막는다. 이 전략에 큰 기대를 걸지 않는것이 좋다.
- 보안관련자료는 공개적으로 리용하여야 할 자료와 분리시키는것이 좋다. Unix 에서 /etc/passwd 는 2가지 자료형을 다 포함한다. 그림자통과암호파일은 요구되는 분리를 보장한다.

제5절. 단번서명

통과암호는 수백년동안 적아를 구분하는데 리용되였다. IT환경에서는 통과암호가 콤퓨터, 콤퓨터망, 프로그람, 파일 등에 대한 접근을 조종한다. 사용자가 어떤 정보를 얻기 위해 싸이버(cyber)공간을 항행할 때 통과암호를 반복하여 입력해야 한다면 그것이 별로 편리하다고 생각하지 않을것이다. 자기의 워크스테이션에 앉아서 콤퓨터망에 있는 봉사기의 자료기지로부터 어떤 정보를 요구할 때 다음과 같이 해야 한다면 좋겠는가?

- 워크스테이션에서 첫번째 통과암호를 입력한다.
- 콤퓨터망에 들어 가기 위해 두번째 통과암호를 입력한다.
- 봉사기에 접근하기 위해 세번째 통과암호를 입력한다.
- 자료기지관리체계에 접근하기 위해 네번째 통과암호를 입력한다.
- 자료기지에 있는 표를 열기 위해 다섯번째 통과암호를 입력한다.

다섯개의 서로 다른 통과암호들을 잠재적으로 기억하고 매 경우에 적당한 하나를 선택해야 하는데 그것을 잊고 같은 통과암호를 5번 반복입력하는것은 매우 나쁜 결과를 초래한다.

단번서명봉사는 이 문제를 해결한다. 사용자가 자기의 통과암호를 한번 입력하면 체계는 이 통과암호를 기억하고 사용자를 다시 인증해야 할 때마다 그를 대신해서 그 일을 하게 된다.

이러한 단번서명봉사는 사용자에게 편리하지만 새로운 보안문제를 만든다. 기억된 통과암호를 어떻게 보호하겠는가? 지금은 체계가 평문(암호화되지 않은 문)으로 통과암호를 요구하기때문에 앞에서 본 일부 기술들을 쓸수 없다.

체계설계가들은 편리성과 보안의 균형을 조절해야 한다. 리용상 편리성은 IT 체계를 실제로 쓸모 있게 만드는데서 중요한 인자이다. 그런데 이렇게 하면 공격 받기 쉽게 된다.

제6절. 그밖의 방법들

만일 통과암호에 의한 보안으로 만족되지 않는다면 무엇을 더 할수 있겠는가? 일반적견해로부터 다음과 같은것들을 선택할수 있다. 사용자는 다음과 같은것들에 기초하여 인증을 받을수 있다.

- 사용자가 알고 있는것.
- 사용자가 가지고 있는것.
- 사용자가 누구인가?
- 사용자가 무엇을 하는가?
- 사용자가 어디에 있는가?

사용자가 알고 있는것

사용자가 인증을 받기 위해서는 몇가지 《비밀》을 알고 있어야 한다. 앞에서 이러한 인증방식의 첫번째 실례를 보았다. 통과암호가 바로 《사용자가 알고 있는것》이다. 다른 하나의 실례는 은행카드나 기타 류사한 통표들에서 리용되는 개인식별번호(PIN)이다. 세번째 실례로 사용자가 자기의 은행구좌에 대하여 전화로 문의할 때의 상황을 고려해 보자. 사용자의 호출을 취급하는 은행사무원은 어떤 정보를 알려 주기전에 사용자에게 집주소, 생일, 처의 이름과 같은 개인정보들을 요구할수 있다.

이와 같은 인증방식에서는 사용자의 비밀자료를 얻는 사람은 바로 《사용자자신이다》. 한편 사용자가 자기의 비밀을 다른 사람에게 넘겨 준다 해도 흔적이 남지 않는다. 다음의 경우를 생각해 보자. 당신이 속한 조직안에서 누군가가 당신의 사용자이름과 통과암호로 가입하여 콤퓨터를 악용한 경우에 당신은 자신의 청백함을 증명할수 있는가? 당신은 자기의 통과암호를 루설하지 않았다는것을 증명할수 있는가?

사용자가 가지고 있는것

사용자는 인증을 받기 위해 물리적통표를 제출하여야 한다. 즉 자물쇠를 여는 열쇠는 사용자가 가지고 있다. 회사의 재산에 대한 접근을 통제하는데 리용되는 카드나 신분증은 통표의 다른 실례들이다. 앞으로 지능카드읽기장치가 워크스테이션들의 표준장비로 된다면 지능카드가 통과암호를 대신하게 될것이다.

물리적통표는 잃어 버리거나 도적 맞힐수 있다. 이전에는 통표를 소유한 사람은 누구나 합법적소유자로서의 동등한 권한을 가지였다. 보안성을 높이기 위해 물리적통표는 흔히 사용자가 알고 있는것(례로 은행카드에서 PIN)과 배합하여 리용하였거나 또는 통표에 합법적사용자를 확인하는 정보(례로 은행카드에 있는 사진)를 포함시켰다. 그러나 이와 같은것으로써는 사기군들이 합법적사용자로 가장하기 위하여 필요한 정보들을 얻는것을 막을수 없으며 또 사용자가 그러한 정보를 본의 아니게 넘겨 주는것을 막을수 없다.

사용자가 누구인가

실제로 사람들을 개별적으로 인증하여야 한다면 생체공학적수법을 사용할수 있다. 사진카드는 이미 언급하였다. 보다 정밀한 방법들에서는 손바닥지문, 손가락지문, 홍채모양, 망막모양 등을 사람을 인식하는데 리용한다.

통과암호에 의한 인증은 매개 인증시도를 명백히 거부하거나 접수한다. 생체공학적 수법에서는 실제 측정한 모양을 기억된 모양과 비교하는데 이것을 엄밀하게 맞추기는 쉽지 않다. 이로부터 틀린긍정과 틀린부정이라는 새로운 문제에 직면하게 된다. 틀린 사용자를 접수하는것(틀린긍정)은 명백히 보안문제이다. 합법적사용자를 거부하는것(틀린부정)은 혼란과 비능률적인 작업환경을 조성한다. 또 다른 문제들도 있다. 사용자들이 이러한 기구들을 받아 들이겠는가? 그들은 자기의 손가락지문이 채취된다면 자기들이 범죄자처럼 취급된다고 느낄수 있다. 또한 자기들의 망막을 주사하는 레이자빛을 좋아 하지 않을수 있다.

사용자가 무엇을 하는가

흔히 사람들은 어떤 개인에게 특징적인 방식으로 어떤 기계적인 과제를 반복한다. 손으로 쓴 수표를 검사하는것은 한가지 실례이다. 여기서 위조는 그리 어렵지 않다. 보안성을 높이기 위해 사용자는 쓰기속도나 쓰기압력과 같은 속성들을 측정하는 특수한 받치개우에서 수표할수 있다. 건반에서는 건누름속도와 건누름간격이 개별적사용자들을 인증하는데 리용되고 있다. 이처럼 인증체계는 목적하는 응용에서 접수할수 있는 정도로 틀린긍정과 틀린부정을 줄이도록 설치되여야 한다.

사용자가 어디에 있는가

체계는 사용자가 가입할 때 그가 어디에 있는가를 등록자리에 받아 들일수 있다. 일부 조작체계들은 사용자가 정해 진 말단에서 가입할 때에만 접근을 허락한다. 실례로 체계관리자는 오직 조작탁에서만 가입할수 있고 다른 사용자말단에서는 가입할수 없다. 류사하게 사용자는 자기 사무실에 있는 워크스테이션에서만 가입할수 있다. 이 류형의 인증은 이동식 및 분산콤퓨터환경에서 적합하다. 만일 인증에 정확한 기하학적위치를 리용해야 한다면 전지구위치결정체계(GPS)의 봉사를 리용할수 있다. 가입요구가 제기될 때 사용자의 위치를 확인하는것은 후에 그 사용자의 신원을 보증하는데 도움이 될수 있다.

통과암호는 개인을 인증하지 못한다. 인증에 통과했다는것은 다만 사용자가 특수한 비밀을 알고 있다는것을 암시할뿐이다. 정당한 사용자와 그 사용자의 통과암호를 얻어 쥔 침입자를 구별할수 없다.

이 장의 문헌안내

Unix통과암호보안에 대해서는 [104]을 참조하시오. 여기서 대표적인 통과암호선택에 대한 흥미 있는 통계를 볼수 있다. 문헌 [48]에서도 이런 문제들이 취급된다. Windows NT 통과암호기구에 대한 서술은 [63, 139]에 있다. 콤퓨터보안에 대한 모든 책들은 통과암호의 적절한 선택과 통과암호보안의 중요성에 대해 많은 조언을 주고 있다. 인터네트비밀(privacy)에 관한 책 [124]는 관련 있는 쏘프트웨어제품에 대한 지시자들과 함께 자동통과암호생성에 대한 내용들이 들어 있다.

인터네트에는 수많은 통과암호크래커들이 있다. 이러한 프로그람들가운데서 하나를 해석하면 통과암호크래커들의 류형과 그것들이 리용하고 있는 사전의 크기와 기교에 대한 좋은 착상을 얻게 될것이다.

련습문제

1. 자기의 콤퓨터체계에서 통과암호기구를 확인하시오. 통과암호의 길이, 통과암호형식, 통과암호사용기한에 대한 어떤 규칙들이 있는가? 체계에 통과암호들이 어떻게 기억되여 있는가?

2. 사용자가 통과암호를 구성할 때 26개의 영어자모문자만을 허용한다고 하자.
   - 통과암호가 최대로 n개(n=4, 6, 8) 문자길이이고 대소문자구분이 없다면 서로 다른 통과암호를 몇개 만들수 있는가?
   - 우와 같은 조건에서 대소문자구분을 한다면 몇개의 통과암호를 만들수 있는가?

3. 통과암호가 6문자길이를 가지며 대소문자를 포함한 모든 영어자모문자를 쓸수 있다고 하자. 완전탐색공격이 평균 얼마동안 가해 지겠는가?
   - 하나의 통과암호를 검사하는데 1/10s 걸린다면?
   - 하나의 통과암호를 검사하는데 1μs 걸린다면?

4. 길이 n인 통과암호를 만드는데 26개 영어자모문자들만 쓸수 있다고 하자. 또한 같은 통과암호를 대소문자구분이 없는 체계와 대소문자를 구분하는 체계에서 각각 사용한다고 하자. 이때 대소문자를 구분하는 통과암호의 판본(version)을 추측하는데 필요한 시도회수의 웃한계를 구하시오.

5. 프로그람짜기: 입력으로 길이 s인 우연2진초기값(seed)을 취하는 통과암호생성기를 만드시오. 다음련습들에서 값 s=8, 16, 32, 64 를 리용하시오.
   - 서로 다른 사용자들에게 당신의 기구를 써서 통과암호를 생성하게 하고 그들이 같은 통과암호를 선택하게 되는 모든 경우들을 조사하시오.
   - 통과암호를 생성하고 그것을 암호화하시오. 우연초기값에 모든 값들을 적용하여 본래의 통과암호를 밝혀 보시오. 통과암호를 발견하기까지 몇번이나 추측을 해야 하겠는가?

6. 토론: 통과암호는 사용자가 입력하고 콤퓨터가 검사한다. 그러자면 사용자와 콤퓨터사이에 통신로가 있어야 한다. 지금까지 우리는 이 통신로를 추상적으로 생각하고 있었으며 그것이 존재하며 충분히 안전하다고 가정하였다. 이 가정이 정당한것은 어느 때인가? 어느 때 정당하지 않은가?

7. 동시에 여러개의 통과암호를 적용해야 한다면 그것들을 통과암호책에 써둘수 있다. 통과암호책은 바로 통과암호들을 포함하는 하나의 보호된 파일이다. 통과암호책에 대한 접근은 다시 기본통과암호를 통해 조종할수 있다. 이러한 기구의 우점은 무엇인가?

8. 문헌 [65]에 서술된 통과암호추측에는 시간/기억기의 이률배반관계가 존재한다. N을 가능한 통과암호의 개수라고 하자. N개의 시험적암호화를 리용하는 예비계산단계에서 $N^{2/3}$개의 기입란이 있는 표가 형성된다. 만일 후에 사용자가 어떤 주어 진 암호화된 통과암호를 찾으려면 $N^{2/3}$개의 시험적암호화를 해야 한다. 5bit 문자모임으로부터 길이 6인 통과암호들을 선택할 때 얼마만한 기억공간이 요구되는가? 하나의 시험적암호화가 1ms 걸린다고 하면 통과암호를 찾아 내는데 시간이 얼마나 걸리겠는가?

9. 상업적리용가능한 생체공학적인증체계들의 봉사를 설명하시오. 당신이 창안한 체계들을 사용자들이 얼마나 잘 접수하겠는가?

제3장. 접근조종

사용자는 체계에 가입하면 새로운 파일들을 창조하며 또한 자기의 파일들을 보호하려고 한다. 파일들의 일부는 공동용이고 일부는 제한된 사용자들을 위한것이며 또 일부는 전용일수 있다. 사용자는 자기가 계획한 접근조종방책을 표현하기 위한 언어와 접근조종을 시행할 기구들을 요구한다. 이 장에서는 접근조종에서 쓰이는 어휘들을 소개한다. 제4장에서 전문적인 접근조종방책들을 취급한다.

목적

- 접근조종의 기초모형을 소개한다.
- 몇가지 접근조작들을 보고 용어들의 실제적인 정의를 자기나름으로 해석하는것은 위험하다는것을 인식한다.
- 전문적인 보안방책들에 의존하지 않는 본질적인 접근조종구조들을 제시한다.
- 보안방책들을 표현하는데 자주 리용되는 부분적순서화들과 살창들 그리고 수학적개념들을 정의한다.

제1절. 배 경

접근조종의 세부에 들어 가기전에 지난 수십년간에 걸쳐 콤퓨터체계들과 그 리용분야들을 개발하여 온 과정을 보기로 하자. 콤퓨터체계들은 자료를 처리하고 기억기, 인쇄기 등과 같은 공유자원들에 대한 접근을 조정한다. 그것들은 기밀성보다는 완정성의 리유로 하여 초보적으로 자료와 자원들에 대한 접근조종을 제공하여야 한다. 전통적인 다중사용자조작체계들은 많은 사용자들에게 일반적인 봉사를 제공한다. 이 조작체계들은 단순하고 일반적인 접근조작들을 가지며 그것들이 조종하는 파일들의 의미와는 관련이 없다. 현대탁상형콤퓨터조작체계들은 개별적사용자들의 작업을 지원한다. 여기에는 개별적응용들에서 전용으로 쓰이는 매우 복잡한 접근조작들이 많다. 흔히 사용자들은 프로그람실행의 상세한 낮은 준위까지에는 흥미가 없다. 그들의 높은 준위 보안요구를 낮은 준위보안조종에 맞추는것은 매우 어렵다. 다시말하여 일반용콤퓨터체계로부터 (유연한) 전용콤퓨터체계에로의 이행을 보여 주고 있다. 이 책에 서술된 서로 다른 접근조종모형들과 비교할 때 이 경향을 념두에 두어야 한다.

제2절. 주동체와 객체

접근조종을 론하기 위해 우선 적합한 전문용어를 정의하여야 한다. 《접근》의 근본의미는 몇가지 특정한 접근조작으로 피동인 객체에 접근하는 능동인 주동체가 있다는것을 암시한다. 이때 참조감시기(제5장)가 접근을 허락 또는 거부한다. 그림 3-1에서 보여준 접근조종의 기초모형은 람프손(Lampson)[82]에 의해 제안되였다.
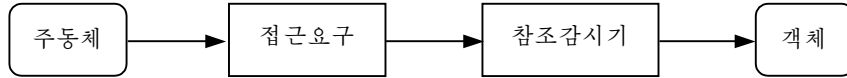
그림 3-1. 접근조종의 기초모형

대표적인 주동체는 사용자 또는 처리(process)이다. 전형적인 객체는 파일이나 기억기, 인쇄기, 콤퓨터망에서의 마디 등과 같은 자원들이다. 그러나 이것은 체계에 있는 매개 실체가 주동체이거나 객체이여야 한다는 식으로 명백한 구분이 있다는것을 의미하는것은 아니다. 정황에 따라서 하나의 실체가 어떤 접근요구에서는 주동체이지만 다른 접근요구에서는 객체일수도 있다. 《주동체》와 《객체》라는 용어들은 하나의 접근요구에서 능동부문과 피동부문을 구별하는데 쓰인다. 접근조종에 초점을 둘 때 주동체와 객체는 다음의 두가지 선택을 가진다.

- 주동체가 무엇을 하도록 허락하겠는가? 혹은
- 객체를 가지고 무엇을 할수 있는가?

이것은 제1장 4절 1의 첫번째 설계원리의 한가지 실례이다. 전통적으로 조작체계의 기본과제는 파일들과 자원들 즉 객체들을 관리하는것이다. 이와 같은 관점에서 볼 때 대체로 뒤의 방법을 취하는 접근조종기구들을 만나게 된다. 그러나 우리는 앞에서 자료기지관리체계와 같은 응용지향IT체계들이 말단사용자들에게 봉사를 제공한다는것을 보았다. 이러한 체계에는 주동체들의 작용을 조종하는 기구들이 적합하다.

제3절. 접근조작

콤퓨터체계를 어떻게 보는가에 따라 접근조작들은 객체지향체계에서의 기본적인 기억기접근으로부터 방법호출에 이르기까지 변한다. 비슷한 체계들이 서로 다른 접근조작들을 리용할수 있으며 지어는 같은것으로 보이는 조작들에 서로 다른 의미를 붙일수도 있다. 여기서는 이 분야에서 중요한 문헌들로부터 전문용어를 채용하여 몇가지 대표적인 접근조작들을 검토해 보기로 한다.

1. 접근방식

주동체는 대다수 요소적인 준위에서 객체를 관찰하거나 변경시킬수 있다. 따라서 다음과 같은 두가지 접근방식을 정의한다.

관찰 : 객체의 내용을 본다.
변경 : 객체의 내용을 변경한다.

대부분의 접근조종방책들을 관찰과 변경이라는 용어로 표현할수 있지만 이러한 방책묘사들은 그것들이 취급하는 응용과 너무 거리가 멀고 정확한 방책이 실현되였는가를 검사하기 어렵게 한다. 따라서 보다 풍부한 접근조작들을 보게 된다.

2. 접근권한과 접근속성

좀더 복잡한 준위에서 우리는 콤퓨터보안의 력사에서 두개의 리정표인 제4장 2절의 벨-라파둘라(Bell-Lapadula)보안모형의 접근권한과 Multics조작체계[119]의 접근속성을 보게 된다.

거기에는 4가지 접근권한 즉 실행, 읽기, 추가(맹목적쓰기라고도 한다), 쓰기가 있다. 그림 3-2는 이 접근권한들과 2개의 기본접근방식인 관찰과 변경사이의 관계를 보여준다.

	실행	추가	읽기	쓰기
관찰			x	x
변경		x		x

그림 3-2. 벨-라파둘라모형에서의 접근권한들

이렇게 정의한 리유를 리해하기 위하여 다중사용자조작체계가 파일들에 대한 접근을 어떻게 조종하는가를 보자. 사용자는 접근이 허락되기전에 파일을 열어야 한다. 보통 파일들은 읽기접근 또는 쓰기접근을 위해서 열려 질수 있다. 이 방법으로 조작체계는 두 사용자가 같은 파일에 동시에 쓰기하는것과 같은 충돌을 피할수 있다. 효과성의 리유로 쓰기접근은 보통 읽기접근을 포함한다. 실례로 어떤 파일을 편집하는 사용자는 한번은 읽기 또 한번은 쓰기를 위해 파일을 두번 열지 않아도 된다. 따라서 쓰기지령을 관찰과 변경을 포함하도록 정의하는것이 좋다.

일부 체계들은 실제로 추가조작을 가지고 있다. 내용을 관찰하지 않고 그 객체를 변경하는것은 흔히 쓰이는 조작은 아니다. 검열기록들은 추가권한을 리용하는 하나의 실례이다. 기록파일에 써넣는 처리는 그 파일에 대한 읽기를 요구하지 않으며 대체로 그것을 전혀 읽지 않는다.
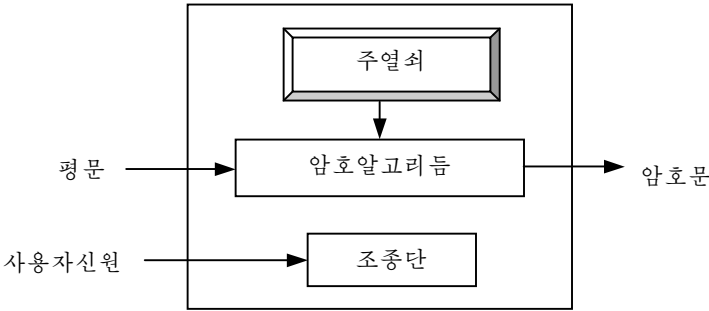
그림 3-3. 암호화엔진

조작체계들은 파일들 례하면 프로그람들을 전혀 열지 않고 리용할수도 있다. 따라서 관찰도 변경도 포함하지 않는 실행권한을 도입하게 된다. 그러면 콤퓨터가 프로그람의

명령들을 읽지 않고 어떻게 실행할수 있는가고 물을수 있다. 물론 그것은 옳으며 Multics의 실행속성은 실제로 실행권한과 읽기권한을 가진다. 그러나 객체의 내용을 읽지 않고 실행에 리용하는 조작들이 있다.

내용을 함부로 고칠수 없는 등록기(tamper-resistant register)에 주열쇠를 보관하는 암호화엔진을 생각하자(그림 3-3).

여기서 물리적으로 주열쇠를 읽어 낼수 있는 방법은 없으나 접근조종규칙들은 이 열쇠를 암호화에 리용하도록 누구를 허락하는가를 통제할수 있다. 여기서 이 열쇠는 읽지 않고 리용할수 있으며 실행권한은 바로 이런 정황에서 설명된다.

누군가가 정의한 접근조작들을 해석할 때 자기나름으로 해석하지 마시오.

Multics조작체계는 자료토막들에 대한 접근속성들과 등록부토막들에 대한 접근속성들을 구별한다. 주어 진 접근권한들의 모임을 객체의 류형에 따라 서로 다르게 해석하는 것은 실제로 공통적인 현실이다. 이러한 접근속성들을 이름 지을 때 또다시 《읽기》나 《쓰기》와 같은 용어들을 리용하게 된다. 표현을 명백히 하기 위해서 벨-라파둘라의 접근권한들을 e, r, a, w 로 표기한다. 그림 3-4는 접근속성들과 접근권한들사이의 대응관계를 보여 준다.

자료토막들		등록부토막들	
읽기	r	상태	r
실행	e,r	상태, 변경	w
읽기, 쓰기	w	추가	a
쓰기	a	탐색	e

그림 3-4. Multics에서의 접근속성들

3. Unix

보다 최근의 실례는 Unix조작체계에서의 접근조종이다. 여기서 접근조종방책들은 다음과 같은 3가지 조작들로 표현된다.

읽기: 파일로부터 읽는다.
쓰기: 파일에 써넣는다.
실행: 파일(프로그람)을 실행한다.

이 조작들은 벨-라파둘라모형에서와는 다르다. 실례로 Unix는 파일의 등록부에로의 쓰기접근을 조종함으로써 누가 파일들을 창조하거나 지울수 있는가를 조종한다. 다른 조작체계들은 이 목적을 위해 특별히 지우기조작을 가진다. Unix에서 어떤 파일에 할당된 접근권한은 그것이 속한 등록부에 있는 그 파일의 기입사항을 수정함으로써 변경된다. 다른 조작체계들은 이를 위한 특수한 조작들을 가진다.

4. Windows NT

마지막실례로 접근조종의 기초로서 Windows NT조작체계의 New Technology File System(NTFS)에서 리용된 허가에 대하여 보자[63]. 그것들은 다음과 같다.

read(읽기)
write(쓰기)
execute (실행)
delete(지우기)
change pemission(허가변경)
change ownership(소유권변경)

여기서는 파일들의 지우기나 접근권한의 변경을 조종하기 위해 등록부조작들에 의존하지 않는다. 접근권한을 변경시키는 조작들은 보안방책을 설정할 때 리용할 다른 하나의 인자이다. 주동체의 접근권한을 취급하는 조작들은 주동체의 접근권한이 어떤 다른 부분에 의해서 변경될 때에는 허락(grant)과 취소(revoke)로 부르고 주동체자신이 자기의 접근권한을 변경할 때에는 주장(assert)과 거부(deny)로 부른다. 이러한 속성의 조작들은 위임방책(delegation policies)들에서 쓰이는데 여기서는 하나의 주동체가 다른 하나의 주동체를 불러 내고 불러 낸 주동체의 권한들을 설정해 주어야 한다.

제4절. 소유권

앞에서 주동체들이 어떻게 객체들에 접근하는가를 조종하는 방책들을 언급하였다. 이러한 방책들은 다음장에서 론의한다. 또한 누가 그 방책을 책임지는가를 서술해야 한다. 여기에 두가지 근본적인 선택이 있다.

- 자원의 소유자가 누구에게 접근허가를 주는가를 선언한다. 이러한 방책은 접근조종이 소유자의 자유이기때문에 자유(discretionary)접근조종이라고 할수 있다.
- 체계전반의 방책이 누가 접근허가를 가지는가를 선언한다. 명백한 근거로부터 이러한 방책은 위임(manadatory)접근조종이라고 할수 있다.

대부분의 조작체계들은 자원의 소유권개념을 지원하며 접근조종결심을 채택할 때 소유권을 고려한다. 그것들은 자원의 소유권을 재정의하는 조작들을 포함할수도 있다.

앞에서 해설한 자유접근조종과 위임접근조종의 직관적인 설명을 콤퓨터보안에서 널리 리용되는 자유 및 위임접근조종에 대한 정의들과 혼돈하지 말아야 한다. 그러므로 이 용어들은 오렌지부크[112]에 서술된 전문적인 접근조종방책들을 따르기로 한다. 자기나름의 해석을 주의해야 한다는것을 다시한번 강조한다.

제5절. 접근조종구조

다음으로 어떤 접근조작들이 허락되는가를 보자. 접근권한들은 주동체와 객체의 매개 조합에 대하여 개별적으로 정의될수 있다. 수많은 주동체와 객체들에 대하여 볼 때 이러한 구조들은 관리하기 어려우므로 조종의 중간준위를 선택하게 된다. 다음과 같은 표기를 받아 들인다.

- 주동체들의 모임 S
- 객체들의 모임 O
- 접근조작들의 모임 A

1. 접근조종행렬

접근권한들은 조종행렬(표)의 형식으로 아주 간단히 정의된다.

$M=(Mso)_{s \in S,\, o \in O}$ 여기서 $Mso \subset A$

여기서 항목 *Mso*는 주동체 *s*가 객체 *o*에 대해서 수행할수 있는 접근조작들의 모임을 나타낸다. 이 방법은 콤퓨터보안의 초기에 제기된것이다[82].

접근조종행렬은 접근허가행렬이라고도 부른다. 접근조종행렬은 추상적인 개념이며 주동체와 객체의 수가 많거나 또는 주동체와 객체들의 모임이 자주 변하면 직접 실현하기가 그리 쉽지 않다.

벨-라파둘라모형(제4장 2절)은 오렌지부크의 자유접근조종방책을 모형화하기 위하여 접근조종행렬을 리용한다.

그림 3-5는 두 사용자와 3개의 파일에 대한 접근조종행렬의 간단한 실례이다.

	bill.doc	edit.exe	fun.com
Alice	—	{execute}	{execute, read}
Bill	{read, write}	{execute}	{execute, read, write}

그림 3-5. 접근조종행렬

bill.doc는 Bill에 의해 읽혀 지거나 써질수 있으며 Alice는 전혀 접근할수 없다.

edit.exe는 Alice와 Bill에 의해 실행은 될수 있으나 그밖의 접근은 할수 없다.

fun.com은 두 사용자에 의해 실행되거나 읽혀 질수 있으나 Bill만이 그 파일에 쓸수 있다.

2. 자격

앞에서 겨우 하나의 접근조종행렬을 직접 실현하였다. 두개의 명백한 선택안중의 하나의 선택이 있다. 접근권한은 주동체 또는 객체에 주어 질수 있다. 첫째 경우에 매 주동체는 자격(capabilities) 즉 이 주동체의 접근권한을 명시하는 위조할수 없는 통표를 가진다. 이 자격은 접근조종행렬에서 주동체의 행에 대응한다. 앞의 실례에서 자격으로 주어 진 접근권한은 다음과 같다.

Alice의 자격: edit.exe: 실행; fun.com: 실행, 읽기
Bill의 자격: bill.doc: 읽기, 쓰기; edit.exe: 실행; fun.com: 실행, 읽기, 쓰기

전형적으로 자격들은 자유접근조종과 관련된다. 주동체가 새로운 객체를 창조할 때 다른 주동체들에게 적절한 자격을 허가함으로써 그 객체에 대한 접근을 줄수 있다. 또한 어떤 주동체(처리)가 다른 주동체를 호출할 때 그의 자격 또는 자격의 일부분이 호출된 주동체에 넘겨 질수도 있다.

자격은 새로운 개념이 아니며 또 그것이 보안기구로 널리 리용되는것도 아니다. 왜냐하면 그것이 보안관리의 복잡성을 초래하며 또한 조작체계들의 전통적인 경향이 객체들을 관리하는 방향이라는것과 관련된다.

- 주어 진 객체에 대한 접근허가를 누가 가지고 있는가를 알아 내기는 어렵다.
- 자격을 취소하기는 매우 어렵다. 그러자면 조작체계가 그에 대한 과제를 받아야 하거나 사용자가 자기의 자격들의 리력을 모두 보관해야 한다. 이 문제는 그 자격에 제3자에게로 자신을 넘겨 줄 권한이 있을 때 특별히 어렵게 된다.

그러나 분산체계의 출현은 자격기초접근조종에서 흥미를 불러 일으켰는데 여기서 보안방책은 콤퓨터망에서 마디들사이를 물리적으로 혹은 가상적으로 떠돌아 다니는 사용자들을 취급해야 한다.

자격들을 리용하자고 결심할 때에는 그것들의 보호에 대하여 생각해야 한다. 즉 자격들을 어디에 기억하는가? 만일 자격들이 단일콤퓨터체계내에서만 리용된다면 조작체계에 의한 완정성보호에만 의존할수 있다(제5장). 그러나 자격들이 망우에서 돌아 다닐 때에는 암호학적인 보호도 요구된다(제12장).

3. 접근조종목록

접근조종목록(ACL-Access Control List)은 객체 그자체와 그 객체에 대한 접근권한들을 기억한다. 그러므로 ACL은 접근조종행렬의 렬(column)에 해당하며 누가 주어 진 객체에 접근할수 있는가를 보여 준다. ACL은 오렌지부크 C2클라스[112]의 안전조작체계의 전형적인 내용이다. 앞의 실례에서 ACL의 형태로 주어 진 접근권한들은 다음과 같다.

Bill.doc에 대한 ACL	Bill: 읽기, 쓰기
Edit.exe에 대한 ACL	Alice: 실행; Bill: 실행
Fun.com에 대한 ACL	Alice: 실행, 읽기; Bill: 실행, 읽기, 쓰기

접근권한들의 관리를 개별적주동체들에만 의거하면 오히려 부담이 된다. 그러므로 일반적으로는 사용자들을 그룹으로 묶고 그 그룹의 접근권한들을 끌어 낸다. Unix에서는 파일들에 붙여 진 간단한 ACL들을 볼수 있는데 이것은 객체들의 3개부류 즉 사용자, 그룹 기타에 대한 기본접근방식들을 보여 준다(제6장 4절). ACL들은 객체들에 대한 접근을 관리하도록 만들어 진 조작체계에 적합한 개념이다. 만일 어떤 사용자에게 주어 진 허가들을 취소하기 위해 그 사용자의 허가들을 알아 내려면 모든 ACL들을 힘겹게 탐색해야 한다.

제6절. 중간조종

대규모체계들인 경우에는 접근조종행렬을 만든다 해도 이 행렬로 표현된 보안방책을 관리하는것은 매우 복잡한 과제이다. 특히 이 행렬의 모든 항목들을 요구대로 설정하는 것은 시끄럽고 오유를 범하기 쉬운 일이다. 더우기 주동체들이나 객체들에만 기초한 접근조종은 보안방책의 제한된 범위만을 지원한다. 접근조종결심에 적절히 포함될수 있는 그이상의 정보는 객체에 접근하기 위하여 주동체가 불러 내는 프로그람에 따른다. 이것

은 1960년대 초 캠브리지에서 개발된 Titan조작체계에서의 접근조종에 대한 다음의 설명문에서 볼수 있는것처럼 전혀 새로운것이 아니다[109].

특별히 사용자의 신원뿐아니라(또는 그대신에) 프로그람의 신원을 접근조종결심을 위한 파라메터로 리용할수 있다.

새로운 기술이 반드시 새로운 보안문제를 낳는것은 아니다. 흔히 《새로운》 문제들은 낡은 문제들의 재생이며 그것들에 대한 해결원리들은 이미 알려 져 있다.

1. 그룹들과 부정허가

그룹은 접근조종방책들에 대한 정의를 간단히 하기 위한 수단으로서 이미 언급되였다. 류사한 접근권한들을 가진 사용자들을 그룹으로 묶고 그룹단위로 객체들에 대한 접근허가를 준다. 어떤 보안방책들은 사용자가 오직 하나의 그룹성원으로 될것을 요구하며 다른 보안방책들은 하나이상의 그룹의 성원자격을 가질것을 허락한다. 그림 3-6은 모든 접근허가를 그룹성원자격을 통하여 중재할수 있는 리상적인 세계를 보여 준다.
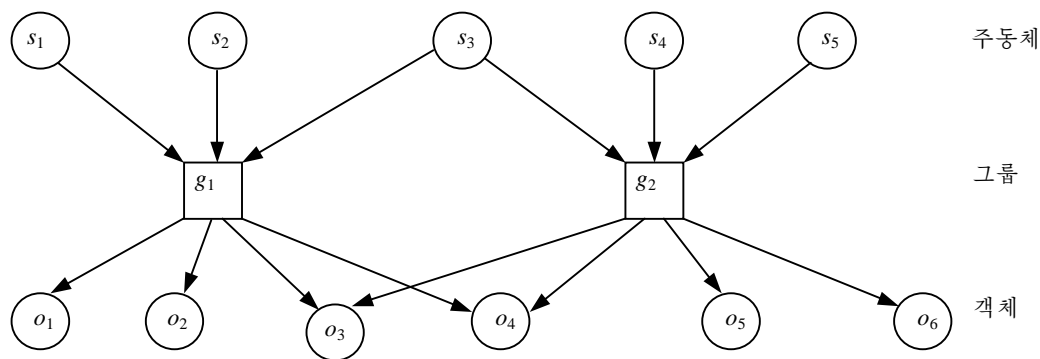
그림 3-6. 그룹들이 접근조종의 중간층으로 된다

흔히 보안방책들에서는 일부 주동체에 하나의 객체에 대한 허가를 직접 주는것 또는 어떤 주동체에 대해 그가 속한 일부 그룹에서 그의 성원자격으로부터 주어 지는 허가를 부인하는것이 편리한 특수경우들이 있다. 부정허가는 접근조종구조에 있는 하나의 기입인데 이것은 주동체가 수행할수 없는 접근조작들을 명기한다. 그림 3-7에서 주동체 s_1는 객체 o_1에 대한 접근이 부인되며 주동체 s_3은 객체 o_5에로의 접근이 허락된다.

2. 보호고리

보호고리들은 주동체와 객체들사이 중간층의 간단한 실례로 된다. 매개 주동체(처리)와 매개 객체는 그의 《중요성》에 대응되는 하나의 번호를 할당 받는다. 대표적실례로 이 번호들을 0, 1, 2, 3이라고 하면 처리들은 다음규칙에 따라서 그 번호들을 접수한다.
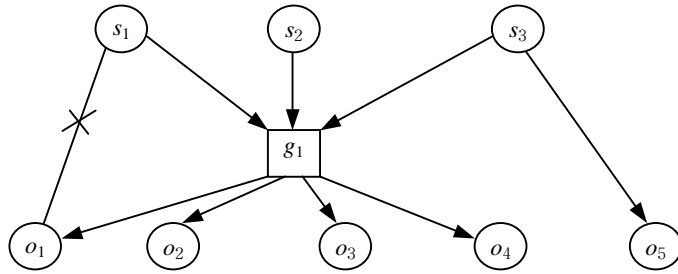
그림 3-7. 부정허가를 가지는 접근조종

0. 조작체계의 핵심부
1. 조작체계
2. 봉사프로그람들
3. 사용자처리들

접근조종결심을 세우기 위해 주동체들과 객체들의 번호를 비교한다(결심의 결과는 보호고리를 리용하여 실현하려는 보안방책에 의존한다). 이 번호들은 동심보호고리들에 대응하는데 중심에 있는 고리 0이 가장 높은 보호준위를 가진다(그림 3-8). 만일 처리가 번호 i로 할당되면 그 처리는 《고리 i에서 실행된다.》고 말한다.
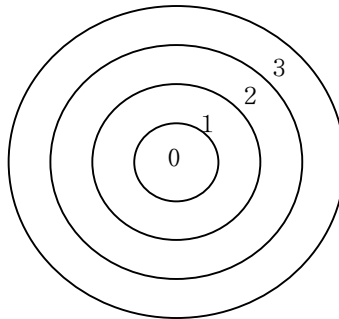
그림 3-8. 보호고리들

보호고리들은 주로 완정성보호에 리용되여 왔다. QNX/Neutrino의 체계/사용자보호는 이러한 보호기구의 최근의 실례로 된다. 여기서는 보호고리들에 다음과 같이 쏘프트웨어요소들을 할당한다.

- Neutrino 마이크로핵심부는 고리 0에서 실행한다.
- Neutrino 처리관리자는 고리 1에서 실행한다.
- 그밖의 모든 프로그람들은 고리 3에서 실행한다.

조작체계코드와 같은 중요한 자료를 포함하는 기억위치들에는 오직 고리 0 또는 1에서 동작하는 처리들만이 접근할수 있다. Unix도 이와 류사한 보호모형을 쓰는데 0준위와 3준위만을 리용한다. 보호고리들은 이미 Multics조작체계에서도 리용되였으며 이와 같은 보안기구를 지원하기 위한 전용하드웨어가 개발되였다[138]. Intel 80386과 80486

처리기들에서도 기계어준위에서 류사한 특징들이 제공된다. 그러나 보호고리들은 일반도구로서 이 책에서와 참고문헌들에서 언급되지 않은 다른 접근조종방책들을 실현하는데도 쓰일수 있다.

3. VSTa 마이크로핵심부에서의 능력

보다 세련된 접근조종방책들을 표현하기 위해서는 보다 유연하고 보다 내적인 구조를 가지는 중간층에 대한 개념이 요구된다. VSTa 마이크로핵심부의 능력(Ability)들은 이러한 개념을 위한 좋은 실례이다. 그것들이 제3장 5절 2에서 정의한것과 같은 완전한 자격들은 아니기때문에 대신에 능력이라는 말을 쓰기로 한다. 능력은 점뒤에 옹근수가 붙은 n개의 옹근수렬형태의 자료구조이다. 례하면 능력은 $.i_1.i_2.\cdots.i_n$으로 표현되는데 여기서 $i_1,\cdots,i_n$은 옹근수들이다. 여기서 길이 n에 대한 제한은 없다. 그리고 n은 0일수도 있다. 능력들에 대한 실례로 .1.2.3이나 .4 또는 .10.0.0.5를 들수 있다. 이러한 내부구조로 하여 능력들의 모임에는 부분순서화(partial ordering)가 존재한다.

정의: 모임 L에 대한 부분순서화 ≤ 은 $L\times L$에서 하나의 관계이며 이것은
반사성 :모든 $a\in L$ 에 대하여 $a\le a$가 성립한다.
이동성 :모든 $a,b,c\in L$ 에 대해서 $a\le b$ 이고 $b\le c$ 이면 $a\le c$이다.
반대칭성 :모든 $a,b\in L$ 에 대해서 $a\le b$ 이고 $b\le a$ 이면 $a=b$ 이다.
만일 두 요소 $a,b\in L$ 를 비교할수 없다면 $a\not\le b$로 쓴다.

능력들은 앞붙이관계를 통하여 순서화될수 있다.

하나의 능력 a_3이 있어서 $a_1=a_2a_3$이라고 쓸수 있다면 능력 a_2은 능력 a_1의 앞붙이이다. 이 경우에 $a_2\le a_1$로 쓴다.

이 앞붙이순서화로써 능력들을 비교할수 있다. 즉 .1≤.1.2≤.1.2.3이지만 .1$\not\le$.4이다. 접근조종방책은 능력들을 가지고 주동체들과 객체들을 표식할수 있으며 주동체의 능력이 객체능력의 앞붙이이면 접근을 허락한다. 이 경우에 모든 객체들에 접근할수 있는 특권사용자의 능력은 빈 문자렬 ε이다. 따라서 어떤 주동체에 능력을 할당하지 않음으로써 그 주동체가 모든 객체에 접근하는것을 허락하게 된다.

주동체와 객체의 속성들을 비교한다. 이러한 속성들중의 어느 하나가 빠지면 무슨 일이 발생하였는가를 늘 검사하여야 한다. 장애안전작용은 접근을 부인하여야 한다는것을 암시한다.

4. 특권들

조작들에 주의를 돌리면 어떤 특권조작들을 실행할 권한을 수집할수 있다. 대표적으로 특권들은 조작체계기능들과 관련되며 체계관리, 여벌만들기, 우편접근, 망접근과 같은 작용들에 관계된다. 특권들은 주동체들과 조작들사이의 중간층으로 볼수 있다(그림 3-9).
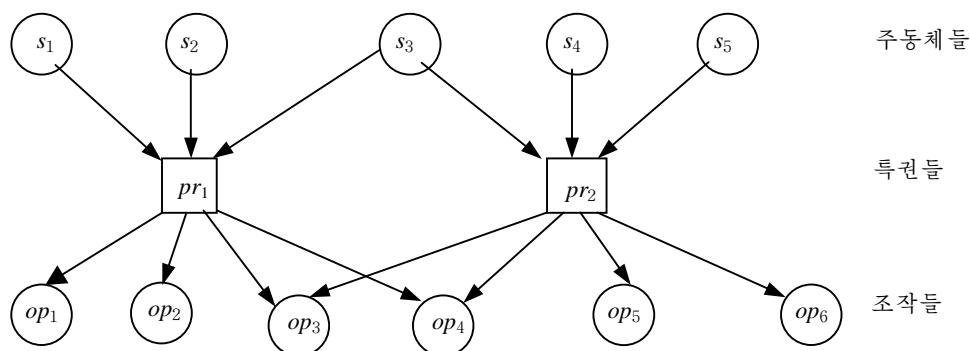
그림 3-9. 주동체와 조작들사이 중간층으로서의 특권들

5. 역할에 기초한 접근조종

특권들은 보통 조작체계에 미리 정해 져 있다. 전용조작들(수속)의 집합을 역할(role)이라고 한다. 주동체들은 그들이 수행하는 역할로부터 자기들의 접근권한들을 끌어 낸다. 역할에 기초한 접근조종(RBAC)은 사용자들과 사용자들이 수행하는 일감들에 초점을 둔다.

주동체들과 객체들사이의 중간층은 접근조종관리의 복잡성을 줄이는데 리용된다. 중간층들은 하나이상의 위치들에 삽입할수 있으며 접근조종을 조직하는데 하나이상의 층을 리용할수 있다.

층들의 선택은 다음과 같은것들을 포함한다.

- **역할**: 역할은 수속들의 집합이다. 역할은 사용자들에게 할당된다. 사용자는 하나이상의 역할을 가질수 있으며 하나이상의 사용자가 같은 역할을 가질수 있다 [134].
- **수속**: 수속은 읽기나 쓰기보다 더 복잡한 의미를 가지는 고준위접근조종방식이다. 수속은 일정한 자료형을 가진 객체들에만 적용될수 있다. 실례로 은행구좌들사이에서의 자금전송을 들수 있다.
- **자료형**: 매개 객체는 일정한 자료형을 가지며 이 자료형에 대하여 정의된 수속들을 통해서만 접근될수 있다. 어떤 객체에 접근할수 있는 수속들을 제한함으로써 그에 대한 접근을 조종하는것이 일반프로그람작성에서 관례로 되여 있다. 이것은 추상자료형의 리론에서 근본적인 개념이다.

이와 같은 구조화된 접근조종은 많은 응용들에서 절실히 요구되지만 아직은 많은 조작체계들에 적용되지 못하고 있다. IBM의 AS/400 [112] 에서의 사용자프로필들과 Windows NT (제7장)에서의 전역그룹와 국부그룹들은 주목할만한 례외이다. RBAC는 자료기지관리체계들에서 많이 쓰인다.

제7절. 보안준위의 살창

콤퓨터보안의 본질적인 사실들을 파악하기 위해서 살창(lattice)에 대한 리해가 필수적인것은 아니다. 그러나 보안문제에 관한 론문들을 읽을 때 살창을 리해하면 도움이 된다. 보안준위는 보호고리나 VSTa능력들과 같은 또 하나의 보안속성이며 보안방책들을 표현하기 위한 기초로서 주동체와 객체들을 표식하는데 리용된다. 간단한 실례로 그림 3-10에 보여 준것처럼 선형으로 순서화된 4개의 보안준위들 즉 《비밀이 아닌》, 《비밀에 관계되는》, 《비밀》, 《극비밀》을 생각해 보자.

오렌지부크의 위임접근조종방책(MAC)과 여러준위보안방책들은 보안준위들에 귀착된다. 이제부터 보안준위들의 모임을 L로 표시하겠다.

만일 보안준위들의 선형적인 순서화를 주장한다면 보안방책들의 제한된 모임밖에 표현할수 없다. 그러므로 보다 일반적인 순서화구조가 있으면 좋을것이다. 제3장 6절 3에서 도입한 보안준위들의 부분순서화 $\leq$ 는 L의 모든 두 요소가 서로 비교할수 있어야 한다는것을 요구하지 않으므로 일부 요구들에 맞는다.

극비밀(top secret)
↑
비밀 (secret)
↑
비밀에 관계되는(confidential)
↑
비밀이 아닌 (unclassified)
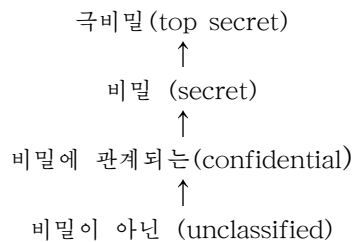
그림 3-10. 선형순서의 보안준위들

다음요구는 주동체의 보안준위가 객체의 보안준위보다 높을 때에만 주동체가 객체를 관찰할수 있게 하는 표준기밀성방책으로부터 나온다. 다음의 두 질문에 유일한 답이 있어야 한다.

- 서로 다른 보안준위에 있는 두개의 객체가 주어 졌을 때 한 주동체가 두개의 객체를 다 읽을수 있게 하는 최소보안준위는 무엇인가?
- 우와 같은 조건에서 한개의 객체가 두개의 주동체에게 읽혀 질수 있게 하는 최대보안준위는 무엇인가?

이 두 질문에 대답하기 위한 수학적구조는 이미 존재한다. 그것을 살창(lattice)이라고 한다. 형식상 그것을 다음과 같이 정의할수 있다.

정의: 살창$(L, \le)$은 모임 L과 부분순서화 $\le$ 로 구성되며 모든 두개 원소 $a, b \in L$에 대하여 가장 작은 웃한계 $u \in L$과 가장 큰 아래한계 $l \in L$이 존재한다. 즉

$a \le u$, $b \le u$이고 모든 $v \in L$에 대하여 $(a \le v \wedge b \le v) \Rightarrow (u \le v)$.

$l \le a$, $l \le b$ 이고 모든 $k \in L$에 대하여 $(k \le a \wedge k \le b) \Rightarrow (k \le l)$.

보안에서는 $a \le b$이면 《a는 b에 의하여 지배된다.》 또는 《b는 a를 지배한다.》고 말한다. 다른 모든 준위들에 의하여 지배되는 보안준위를 《체계낮음》(System Low)라고 한다. 또 다른 모든 준위를 지배하는 보안준위를 《체계높음》(System High)라고 한다.

살창의 대표적인 실례를 그림 3-11에 보여 준다. 모임 L은 $\{a, b, c\}$의 Power모임 $P(\{a, b, c\})$이다. 부분순서화는 부분모임관계 $\subset$이다. System Low는 빈모임 ϕ를, 그리고 System High는 모임 $\{a,b,c\}$를 가진다. 부분모임관계는 그라프로 묘사되며 여기서 마디들은 $P(\{a,b,c\})$의 요소들이다. 마디들사이의 화살표는 부분순서화의 《골격》(skeleton)을 준다. 엄밀하게 $A, B \in P(\{a,b,c\})$에 대하여 만일 $A \subset B$이고 $A \ne B$이면 또 그때에 한하여 A로부터 B에로의 화살표를 그을수 있으며 그리고 $A \subset C \subset B$ 이고 $A \ne C$, $B \ne$C이면 $C \in P(\{a,b,c\})$가 존재하지 않는다.
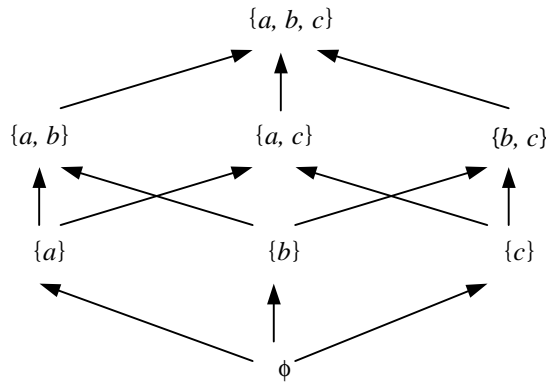
그림 3-11. 살창$(P(\{a, b, c\}), c)$

이러한 규칙에 따라 A로부터 B에로의 화살들의 련쇄가 있으면 A는 B의 부분모임이다. 이 책에서는 부분순서화를 시각화하는데 이러한 수법들을 사용한다.

살창의 다음실례로 오렌지부크와 기타 많은 문헌들의 여러준위보안방책에 널리 쓰이는 한가지 구성을 소개한다. 구체적인것은 제4장에서 취급한다.

- 계층(선형)순서화 $\le_H$를 가지는 분류(classification)들의 모임 H를 취한다.

- 범주(category)들 례하면 대상과제(project)들의 이름, 회사들, 학문분야들 등의 모임 C를 취한다.
- 구획(compartment)은 범주(category)들의 모임이다.
- 보안표식(securiy label)(보안준위)은 쌍(h,c)이며 여기서 $h \in H$는 보안준위이고 $c \subset C$는 구획이다.
- 보안표식들의 부분순서화 ≤는 만일 $h_1 \leq_H h_2$이고 $c_1 \subset c_2$ 이면 또 그때에 한하여 $(h_1, c_1,) \leq (h_2, c_2)$에 의해 정의된다.

그림 3-12가 이 구조를 보여 준다. 두개의 계층준위 즉 공개(public)와 비공개(private)이 있고 두개의 부분 PERSONNEL과 ENGINEERING이 있다.

이로부터 얻어 진 살창에서 다음관계가 성립한다.

(public, {PERSONNEL})≤(private, {PERSONNEL}),

(public, {PERSONNEL})≤(public, {PERSONNEL, ENGINEERING}),

(public, {PERSONNEL}) ≰ (private, { ENGINEERING}).

이러한 보안표식들의 살창을 가지고 위임최소특권(need-to-know)방책들을 실현할수 있다. 그것이 어떻게 작용하는가를 보기 위하여 앞에서 본 간단한 기밀성방책의 관점에서 그림 3-12의 살창을 보자. 보안표식 (private, ENGINEERING)을 가진 주동체는 표식에 범주PERSONNEL을 가지는 객체는 읽을수 없다. 즉(public, {PERSONNEL, ENGINEERING})로 표식된 객체조차도 읽을수 없다.
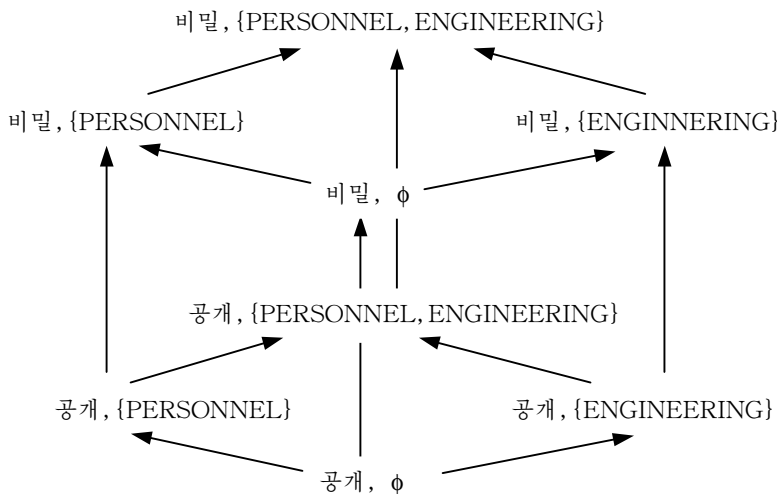
그림 3-12. 보안표식들의 살창

우연히는 아니지만 여기서는 여러준위보안방책들에서 흔히 쓰는 그림 3-10의 간단한 계층살창을 가지고 보안살창들의 론의를 시작하였다. 체계들은 매우 높은 준위의 담보밑에 이 방책들을 실행하도록 구축된다. 다음에 큰 방책변화를 표현할수 있게 구획들을 추가한다. 오늘날 높은 준위 담보때문에 여러준위보안체계를 사용한 응용들을 볼수 있는

데 사실 그것들의 보안준위에는 계층적요소들이 전혀 없다. 실례로 방화벽방법은 망의 내부와 외부사이를 엄격히 분리하기 위하여 그림 3-13과 같은 살창을 리용한다. 이것을 주장하기 위해 부분순서화를 어떻게 해석하는지 알아야 하는것은 아니다.
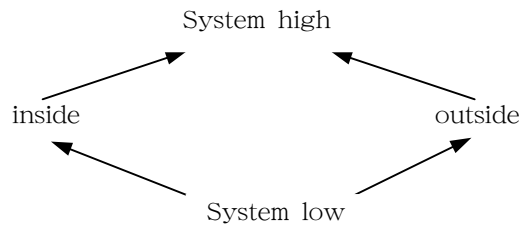
그림 3-13. 방화벽을 위한 살창

이 장의 문헌안내

기본적인 접근조종구조들과 보안살창들은 문헌 [1,39,125]들에서 취급되였다. 접근조종(보호)에 대한 초기론문들은 [82, 131]이고 [158]은 1960년에 개발된 조작체계의 접근조종에 대하여 취급하였다. 보호고리를 위한 보안방책의 실례들은 [111, 125]에 주어졌다. 역할기초접근조종에 대한 최근 자료들은 [134]에서 언급되였다.살창기초접근조종모형들은 [123]에서 볼수 있는데 기밀성과 완정성을 취급하기 위한 그의 응용에 대해서도 서술하고 있다.

QNX/Neutrino 마이크로핵심부와 VSTa마이크로핵심부에 대한 Web페지들은 다음과 같다.

http://www.qnx.com/literature/nto_sysarch/ntosysarch.html
http://www.zendo.com/vsta/vsta_intro.html

련습문제

1. 등록부에 대한 접근조작들을 표현하는데 두 비트가 주어 졌다면 4개 조작들을 어떻게 정의하겠는가? 파일들의 창조와 삭제를 어떻게 조종하겠는가? 이러한 접근조작들을 가지고 숨은(hidden)파일들의 개념을 어떻게 실현하겠는가?(숨은파일들은 권한 있는 주동체들에게만 보인다.)

2. 4가지 접근조작들 즉 읽기(read), 쓰기(write), 허락(grant), 취소(revoke)를 가지는 체계를 생각하자. 허락(grant)은 다른 주동체들에게 읽기와 쓰기접근을 주기 위해서뿐아니라 자기가 소유한 객체들에 대한 접근을 허락하는 권한을 그들에게 넘겨

주기 위해서도 사용할수 있다. 허락과 호출조작을 실현하는데 어떤 자료구조와 알고리듬을 리용하면 자기가 소유한 어떤 객체에 대한 모든 접근을 호출할수 있겠는가?

3. 토론: 그룹과 역할(role)사이의 차이는 무엇인가?

4. 제3장 6절 3에서 정의한 능력들의 부분순서화가 왜 살창을 구성하지 못하는지 설명하시오. 능력들의 모임에 어떤 다른 필요한 요소들을 추가하여 부분순서화를 살창으로 변환하시오.

5. 주동체의 보안준위가 객체의 보안준위를 지배할 때에만 주동체가 객체에 접근할수 있다고 하는 보안방책이 주어 졌다. 이 보안방책과 함께 그림 3-14의 살창을 리용하면 어떤 효과가 있는가?
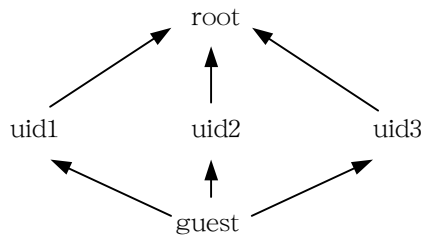
그림 3-14

6. $(L, \leq)$이 보안준위살창이고 여기서 L은 유한모임이라고 하자.이와 같은 살창에 System Low와 System Heih가 유일하게 존재한다는것을 말하시오.

7. 보안준위들이 public, confidential, strictly confidential이고 부문들이 MIN, LECTURERS, STUDENTS로 주어 졌을 때 보안표식들의 살창을 구성하시오. 최소특권(need_to_know)방책에서 보안준위(confidential, {STUDENTS})를 가지는 주동체에게 어떤 객체들이 보이겠는가? n개의 보안준위와 m개의 부문들로부터 얼마나 많은 표식들이 얻어 질수 있는가? 실례로 n=16, m=64인 경우를 생각해 보시오.

8. 보안표식들로서 구획(compartment)들의 살창을 리용하는 보안방책이 주어 졌다. 접근은 주동체의 표식이 객체의 표식의 부분모임일 때에만 허가한다. ADMIN, LECTURERS, STUDENTS라는 부문들을 가지고 어떤 객체들이 표식 {STUDENTS}를 가진 주동체에 의해 접근될수 있는가? 왜 표식 {ADMIN, STUDENTS}를 가지는 주동체가 표식 {STUDENTS}를 가지는 주동체보다 더 속박되는가? 이 방책에서 표식 Φ과 {ADMIN, LECTURERS, STUDENTS}의 역할들을 해석하시오.

9. 부문들의 모임이 주어 졌다. 주동체의 접근권한을 선택적으로 취소시키는 살창기초 need_to_withhold방책을 작성하시오.

제4장. 보안모형

당신의 보안방책은 무엇인가? 누가 당신의 자료에 접근하는가를 어떤 규칙들이 결정하는가? 보안방책을 정식화하기 위하여서 그 방책에 의하여 지배되는 실체들을 서술해야하며 또 방책을 구성하는 규칙들을 서술해야 한다. 보안모형이 바로 그 일을 한다. 이장은 앞에서 학습한데 기초하여 가장 우수한 몇가지 보안모형들을 조사한다.

보안모형들은 기밀성을 위한 방책들(벨-라파둘라)과 완정성을 위한 방책들(비바,클라크-윌슨)을 취한다. 일부 모형들은 정적인(벨-라파둘라)환경에 방책들을 적용하고 기타는 접근권한들의 동적변화를 고려한다(《장성》).

형식적보안모형들(벨-라파둘라와 같은)은 믿을만한 보안평가들에서 중요한 자리를 차지한다. 클라크-윌슨과 같은 비형식적인 모형들은 보다 서술적인 구조로 보안방책들을 나타낸다.

목적

- 보안방책들이 형식적인 수법에 의해 어떻게 표현되는가를 설명한다.
- 콤퓨터보안의 력사에서 중요한 리정표들을 제시한다.
- 여러가지 보안모형들의 유효범위와 한계들을 리해한다.
- 보안에서 일부 결심문제들은 고유하게 결정할수 없다는것을 인정한다.

제1절. 상태기계모형

상태기계(자동체)들은 계산체계들의 여러 측면들을 모형화하는 하나의 공통적인 도구들이다. 이러한 상태기계들은 또한 일련의 중요한 보안모형들을 위한 기초로도 된다. 상태기계모형에서 본질적인것은 상태의 개념과 리산점들에서 일어 나는 상태이행의 개념이다. 상태란 어느 한 순간의 고찰하는 체계에 대한 묘사를 말하는데 이것은 고찰하는 문제와 관계되는 체계의 측면들을 정확히 반영하여야 한다. 가능한 상태이행은 상태이행(다음상태)함수에 의해서 서술되는데 이것은 현재상태와 입력에 따라 다음상태를 정의한다. 출력도 역시 생성할수 있다.

만일 상태기계모형을 리용하여 보안과 같은 체계의 특정한 어떤 속성에 대하여 말하려고 한다면 먼저 이 속성을 만족시키는 모든 상태들을 동일시하여야 한다. 다음에 모든 상태이행들이 이 속성을 유지하는가를 검사하여야 한다. 만일 체계가 이 속성을 가지는 초기상태에서 출발한다면 다음에 그 속성이 항상 유지될수 있는가는 귀납법에 의해 확인할수 있다.

제2절. 벨-라파둘라모형

보안모형들은 보안체계의 설계와 해석에서 중요한 개념이다. 그것들은 체계에서 실시될 보안방책을 표현한다. 벨-라파둘라모형(BLP)은 가장 널리 알려 진 보안모형들중의 하나이다. 그것은 안전한 다중사용자조작체계를 설계하기 위한것으로서 벨(Bell)과 라파둘라(LaPadula)에 의해서 개발되였다. BLP는 접근조종의 기밀성측면들을 취급하는 상태기계모형이다. 접근허가들은 접근조종행렬과 보안준위들을 통하여 정의된다. 보안방책들은 높은 보안준위로부터 낮은 보안준위에로 정보가 내리흐르는것을 막는다. 이러한 방책들은 일반적으로 여러준위보안(MLS)에 귀착된다. BLP는 주동체가 어떤 객체를 관찰하거나 변경시킬 때 일어 나는 정보흐름만을 고찰한다. BLP모형에서는 다음과 같은 모임들을 정의한다.

- 주동체들의 모임 S；
- 객체들의 모임 O；
- 제3장 3절 2의 접근권한들을 직접 반영하는 접근조작들의 모임 A={execute，read，append，write}；
- 부분순서화 ≤ 를 가지는 보안준위들의 모임 L；

어떤 체계의 보안을 검사하기 위하여 그 체계의 상태를 리용한다. 그러므로 모형의 상태모임은 객체들에 접근하는 주동체들의 모든 현재허락들과 현재실례(instance)들을 표현하여야 한다. 이것은 복잡한 상태모임 $B\times M\times F$ 를 만들어 낸다. 여기서

- $B=P(S\times O\times A)$는 현재접근들의 모임이다. 요소 $b\in B$ 는 요소결합(s, o, a)들의 집합이며 주동체 s가 객체 o에 대하여 현재 수행하는 조작 a를 가리킨다.
- M은 접근허가행렬 $M=(M_{so})_{s\in S,\ o\in O}$들의 모임이다.
- $F\subset L^s\times L^s\times L^o$는 보안준위할당들의 모임이다. 요소 $f\in F$는 3중요소 (f_S, f_C, f_O)이며 여기서
    - f_{S}：$S\to L$ 은 매개 주동체가 가질수 있는 최대보안준위를 준다.
    - f_C：$S\to L$ 은 매개 주동체의 현재보안준위를 준다.
    - f_O：$O\to L$ 은 모든 객체들의 (보안)등급을 준다.

주동체의 현재준위는 그의 최대준위보다 높을수 없으며 따라서 $f_C\le f_S$이다. 이것을 다른 말로 《f_S는 f_C를 지배한다.》라고 한다. 최대보안준위를 때때로 주동체의 기밀취급허가(clearance)라고도 한다. 다른 원천들은 기밀취급허가를 다만 사용자들의 보안준위를 표시하기 위해서 리용한다.

상태모임을 잘 정의하는것은 BLP에서 중요한 문제이다. BPL보안속성을 주는데는 입력, 출력 그리고 엄밀한 상태이행의 구조를 서술하지 않아도 된다.

1. 보안방책

BLP는 상태들의 속성으로서 보안을 정의한다. 여러준위보안방책들은 주동체의 보안준위가 객체의 (보안)등급(classification)을 지배할 때에만 주동체가 객체를 읽도록 허

락한다. 이러한 여러준위보안방책들은 위임(mandotory)보안방책이라고 한다. 첫번째 명백한 속성은 단순보안속성(ss-property)이다.

단순보안속성(ss-속성): $(s, o, a) \in b$ 의 매개 요소에 대하여(여기서 접근조작 a는 읽기 또는 쓰기이다.) 주동체 s의 보안준위가 객체 o의 (보안)등급을 지배하면 즉 $fc(c) \le fs(s)$이면 상태 (b, M, f)는 단순보안속성을 만족시킨다.

그러나 단순보안속성은 낮은 준위의 주동체가 높은 준위객체의 내용을 읽는것을 막기에는 충분치 않다. 그것은 높은 준위객체를 읽고 그것을(그의 내용을) 낮은 준위객체에 복사하는 높은 준위트로이목마(Trojan horse)를 만들어 낼수 있다. 그래서 별표속성(*-property)을 통하여 쓰기접근을 조종한다.

별표속성(*-속성): 만일 $(s, o, a) \in b$인 매개 요소에 대하여(여기서 접근조작 a는 추가나 쓰기이다.) 객체 o의 (보안)등급이 주동체 s의 현재준위를 지배하면 즉 $fc(s) \le fo(o)$이면 상태 (b, M, f)는 별표속성을 만족시킨다. 이것은 내려쓰기보안방책이 아니다. 더우기 어떤 요소 $(s, o, a) \in b$(여기서 접근조작 a는 추가 또는 쓰기이다)가 존재하면 $(s, o', a') \in b$와 읽기 또는 쓰기인 a'을 가지는 모든 객체 o'에 대해서 $fo(o') \le fo(o)$를 가져야 한다.

이 정의는 높은 준위주동체가 낮은 준위주동체에 통보문을 보낼수 없다는것을 직접 암시한다. 이 제한을 벗어 나기 위한 두가지 방법이 있다.

- 림시로 높은 준위주동체의 준위를 떨군다. 이것은 현재보안준위 fc를 도입하는 리유이다.
- 별표속성을 위반하도록 허락된 주동체들의 모임을 동일시한다. 이런 주동체들을 신용 받는 주동체들이라고 한다.

첫째 방법은 주동체가 그의 준위가 떨어 질 때 높은 보안준위에서 알고 있던 모든것을 잊어 버린다고 가정한다. 이것은 주동체를 사람처럼 본다면 인정하기 어려운것이지만 BLP는 콤퓨터를 모형화한것이다. 주동체들(처리들)은 자기의 고유한 기억을 가지지 못한다. 그것들이 《아는》것이란 그것들이 관찰하도록 허락된 객체들(파일들)의 내용뿐이다. 이런 상황에서 림시준위를 떨구는것은 문제의 해결책으로 된다.

둘째 방법에서 fs는 사용자의 취급허가(최대보안준위)를 서술한다. 사용자들은 자기들의 최대보안준위아래에서 가입이 허락되며 fc는 사용자가 실제로 가입한 준위를 가리킨다.

둘째 방법을 채용할 때 별표속성은 믿을수 없는 주동체들에 대해서만 취해 져야 한다. 정의에 의하여 신용 받는 주동체는 보안방책을 위반할수 있다. 실제로 자기를 해칠수 있는 체계요소들에 대한 지적자로서 형용사적인 《신용 받는》이라는 단어를 엄밀하게 사용할수 있다. 반대로 주동체가 자신을 해치지 않는다는것을 담보한다면 그것을 믿을만 하다고 한다(turstworthy).

오렌지부크는 이름 붙은 사용자들과 이름 붙은 객체들에 기초한 접근을 조종하는 방책에 대하여 자유접근조종(DAC)이라는 용어를 쓰고있다. 접근허가를 유지하는 주동체들은 그 허가를 다른 주동체들에게 넘겨 줄수 있다. BLP에서 이러한 방책들은 접근조종행렬에 의해 표현되며 다음과 같은 자유보안속성에 포함된다.

자유보안속성(ds-property): $(s, o, a) \in b$의 매개 원소에 대하여 $a \in M_{so}$를 가지면 상태(b, M, f)는 자유보안속성을 만족시킨다.

이상의 3가지 보안속성이 모두 만족되면 상태는 안전하다고 말한다.

2. 기본보안정리

상태 $v_1=(b_1, M_1, f_1)$와 상태 $v_2=(b_2, M_2, f_2)$이 둘 다 안전하면 상태 v_1로부터 상태 v_2에로의 이행은 안전하다고 말한다. 새로운 상태가 안전하다는것을 어떻게 검사할수 있는가를 보기 위해 단순보안속성을 실례로 들자.

다음과 같을 때에만 상태이행은 단순속성을 보존한다.

1. 매개 $(s, o, a) \in b_2 \setminus b_1$는 f_2에 관한 단순보안속성을 만족시킨다($b_2 \setminus b_1$는 b_2와 b_1사이의 모임차를 나타낸다).
2. $(s, o, a) \in b_1$가 f_2에 관한 단순보안속성을 만족시키지 않는다면 $(s, o, a) \notin b_2$이다.

별표보안속성과 자유보안속성의 보존은 같은 방법으로 서술할수 있다. 이제 BLP모형의 중요한 속성을 서술하자.

기본보안정리 : 어떤 체계의 상태이동이 모두 안전하고 이 체계의 초기상태가 안전하면 뒤따르는 모든 상태도 어떤 입력이 일어 나든 안전하다.

이 정리에 대한 형식적증명은 입력렬에 대한 귀납법으로 진행할수 있다. 증명은 매개 상태이행이 보안을 보존한다는 사실우에서 진행되며 특정한 BLP보안속성에 관계되지 않는다.

기본보안정리는 BLP 모형에서 선택된 특정한 보안속성들의 결과물이 아니라 상태기계모형화의 인위적결과이다.

실천에서 기본보안정리는 체계의 보안을 증명하는데 쓰인다. 보안을 보존한다는것을 보기 위해 매 상태이행을 각각 검사하여 안전한 초기상태를 확인해야 한다. 체계가 이 안전한 초기상태에서 기동하는 한 그것은 안전하게 유지될것이다.

3. 안정

1987년에 맥클린(McLean)이 론문[96]에서 다음과 같은 상태이행을 포함한 체계를 제안함으로써 BLP모형의 가치에 대한 론쟁이 벌어 졌다.

- 모든 주동체들을 가장 낮은 보안준위로 준위를 떨군다.
- 모든 객체들을 가장 낮은 보안준위로 준위를 떨군다.
- 접근조종행렬 M의 모든 위치들에 모든 접근권한들을 기입한다.

이 이행에 의하여 도달된 상태는 BLP의 정의에 의하여 안전하다. 이 상태가 실제로 안전한가? BLP가 그렇다고 말할 때 BLP가 보안을 정확하게 취하는가? 여기에는 두가지 의견이 있다.

- BLP를 반대하는 경우(Mclean): 직관적으로 볼 때 모든 사람이 모든것을 읽을수 있게 허락하는 상태로 이행할수 있는 체계는 안전하지 않다. 그러므로 BLP를 개조하여야 한다.
- BLP를 지지하는 경우(Bell): 사용자들이 이와 같은 상태이행을 요구한다면 그것은 보안모형에서 허락되여야 한다. 그것이 요구되지 않는다면 실현하지 않아도 된다. 이것은 BLP의 문제가 아니라 보안요구를 정확히 포착하는 문제이다.

이와 같은 의견불일치의 원인으로 되는것은 접근권한을 변경시키는 상태이행이다. 이러한 이행들은 BLP의 일반틀거리내에서 확실히 가능하다. 그러나 모형의 창시자들은 사실 접근권한이 고정되는 체계를 예상하고 있었다. 보안준위와 접근권한들이 절대로 변하지 않는 속성을 바로 안정(tranquility)이라고 한다.

4. BLP의 측면들과 한계

BLP는 매우 의의 있는 보안모형이다. 그것은 안전한 조작체계들의 설계에서 중요한 역할을 하였으며 때문에 어떤 새로운 모형이 나오면 BLP와 비교한다. 여기서는 BLP의 몇가지 특징들을 보기로 한다.

1. 모형의 서술능력: BLP 상태모임은 모든 현재접근조작들과 모든 현재접근허가들을 서술한다.
2. 보안방책들은 보안준위들과 접근조종행렬에 기초한다. 여기에 다른 구조들을 도입하기는 쉽다. 실례로 어떤 주동체가 일정한 프로그람들을 통해서만 객체들을 접근하게 하는 접근조종을 모형화하기 위해서는 $S \times S \times O$접근조종구조가 보다 적합하다.
3. 실제적보안속성들: BLP에서는 ss-속성, *-속성, ds-속성을 가진다. 비바모형(제4장 5절)은 그의 보안속성들이 기본적으로 BLP와 다르다.
4. 특수한 풀이: 실례로 Multics해석(제5장 4절 3)에서의 상태이행들.

BLP는 접근조종에 의하여 보안을 정의하므로 통속적이라고 볼수 있다.

그러므로 어떤 조작체계나 자료기지관리체계의 작용을 BLP에 의하여 표현하는것은 그리 어렵지 않다. 그러나 BLP가 중요한 보안모형이기는 하지만 보안의 모든 측면을 다 포함하지는 못한다. 그것은 다음과 같다.

- 기밀성만을 취급하며 완정성을 취급하지 못한다.
- 접근조종의 관리를 취급하지 못한다.
- 잠복통로들을 포함한다.

완정성방책들이 없다는것은 BLP의 결함인것이 아니라 BLP의 특징이다. BLP는 접근권한들을 수정하는 방책들을 가지지 않는다. 사실상 BLP의 목적은 본래 보안준위의 변화가 없는 체계들이였다.

잠복통로란 보안기구에 의해 조종되지 않는 정보흐름을 말한다[150]. 만일 낮은 준위주동체들이 높은 준위객체에 대해서 이름들은 볼수 있고 내용에 대해 접근할수 없다면 이때 객체이름들은 분명히 잠복통로이다. BLP에서는 접근조종기구 그자체를 잠복통로를 구성하는데 리용할수 있다. 다음과 같이 정보는 높은 보안준위로부터 낮은 보안준위에로 흐를수 있다.

- 낮은 준위주동체가 자기의 준위에 dummy.obj 라는 객체를 창조한다.

- 그의 높은 준위공범자(트로이목마)가 dummy.obj의 보안준위를 높은 준위로 올려놓거나 또는 그대로 둔다.
- 후에 낮은 준위주동체는 dummy.obj를 읽으려 한다. 이 요구가 접수되는가 거부되는가에 따라 높은 준위주동체의 작용을 알수 있다. 즉 일련의 정보가 높은데로부터 낮은데로 흐른것이다.

이처럼 어떤 조작이 허락되지 않는 주동체에게 접근하려고 하면 일정한 정보흐름이 형성된다. 이것은 자료기지보안에서 흥미 있는 결과를 준다(polyinstantiation-다중구체례제시). 이런 류형의 문제를 피하기 위하여 하나의 객체가 서로 다른 보안준위에서 서로 다른 값을 가지게 할수 있다.

때때로 객체의 내용만을 숨기는것으로는 안심할수 없다. 그의 존재까지도 숨겨야 할수 있다.

제3절. 해리슨-루조-울만모형

벨-라파둘라모형은 접근권한들을 변경시키거나 주동체와 객체들을 창조하거나 삭제하는 방책들을 표현할수 없다. 해리슨-루조-울만(Harrison-Ruzzo-Ullman)모형(HRU)은 이 문제를 취급하는 권한부여체계를 정의한다[64]. HRU모형을 서술하기 위해서는 다음과 같은 모임들이 요구된다.

- 주동체들의 모임 S,
- 객체들의 모임 O,
- 접근권한들의 모임 R,
- 접근행렬 $M=(Mso)s\in S,\ o\in O$; 여기서 항목 Mso 는 주동체 s가 객체 o에 대해서 가지는 권한들을 나타내는 R의 부분모임이다.

주동체모임, 객체모임, 접근행렬을 처리하기 위한 6가지 기본조작들이 있다.

enter r into Mso
delete r from Mso
create subject s
delete subject s
create object o
delete object o

HRU모형에서 지령들은 다음과 같은 형식을 가진다.

Command $c(x_1, x_2, \cdots, x_k)$
if r_1 in Ms_1, o_1 and
if r_2 in Ms_2, o_2 and
… … …

if r_m in Ms_m, o_m and

then

op_1
op_2
.
.
.
op_n

end

첨수들인 $s_1, \cdots$, s_m, 과 o_1, $\cdots$, o_m 은 파라메터목록(x_1, $\cdots$, x_k)에 나타나는 주동체와 객체들이다. 조건들은 특별한 접근권한이 있는가를 검사한다. 조건들의 목록은 비여있을수도 있다. 만일 모든 조건이 맞으면 기본조작들을 순차적으로 실행한다. 매개 지령은 적어도 하나의 조작을 포함한다. 실례로 지령

command *create_file*(*s. f*)
create *f*
enter o into M_{sf}
enter r into M_{sf}
enter w into M_{cf}
end

은 주동체s가 새로운 파일 f를 창조하기 위해서 리용하는데 s는 그 파일의 소유자(접근권한 o)로서 그 파일에 대한 읽기, 쓰기허락을 가진다(접근권한 r와 w). 파일 f의 소유자 s는 다음과 같은 지령으로 읽기권한을 다른 주동체 P에 넘겨 준다.

command *grant_read*(*s*, *p*, *f*)
if o in M_{sf}
then enter r in M_{pf}
end

지령의 효과는 접근행렬의 변화로서 기록된다. 그러므로 접근행렬은 체계의 상태를 서술한다. 변경된 접근조종행렬은 M' 로 표시된다. HRU모형은 접근권한들의 할당을 규제하는 보안방책을 표현할수 있다. 체계가 이러한 방책에 따르다는것을 확인하기 위해 바라지 않는 접근권한들을 허락하는 방법이 존재하지 않는가를 검사하여야 한다.

정의: 만일 접근행렬에 포함되지 않은 권한 r를 그 행렬의 어떤 위치에 첨가하는 지령c가 존재한다면 상태 레하면 접근행렬 M은 권한 r를 루설한다고 말한다. 보다 형식화하면 s와 o가 존재하여 $r \notin M_{s,o}$이고 $r \in M'_{s,o}$이다.

정의: 만일 r 를 루설하는 어떤 상태에로 M을 넘길수 있는 지령렬이 존재하지 않으면 그 상태 레하면 접근행렬 M은 권한 r에 관하여 안전하다고 말한다.

HRU모형의 보안방책을 확인하는것은 이와 같이 안전속성들을 확인하는것으로 된다. 그러나 이제 자신이 불리한 위치에 있다는것을 알게 될것이다.

정리: 접근행렬 M과 권한 r가 주어 졌을 때 권한 r에 관하여 M의 안전성을 증명하는 것은 결정할수 없는 문제이다.

이처럼 안전성문제를 완전한 일반성을 띠도록 할수는 없으므로 모형이 성공하도록 제한하여야 한다. 실례로 하나의 조작만을 포함하는 지령들 즉 단일조작지령들만을 허락하게 할수 있다.

정리: 단일조작권한부여체계와 접근행렬 M 그리고 권한 r가 주어 졌을 때 권한 r에 관하여 M의 안전성을 증명하는것은 결정가능하다.

여기서 안전성문제를 또다시 결정불가능으로 만들기 위해서는 한개 지령당 두개이상의 조작을 허락하면 된다. 안전성문제를 다루기 쉽게 할수 있는 또 하나의 방법은 권한부여체계의 크기를 제한하는것이다.

정리: 주동체의 수가 유한이라면 임의의 권한부여체계의 안전성문제는 결정가능하다.

안전성문제의 결정가능성에 대한 이러한 결과들은 3번째 설계원리의 륜곽을 보여 준다. 만일 복잡한 모형들로만 묘사될수 있는 복잡한 체계들을 설계한다면 보안성을 증명하기 어렵게 된다. 최악의 경우(결정불가능성) 모든 경우들에 대해서 보안성을 증명하는 만능적인 알고리듬은 존재하지 않는다. 만일 증명가능한 보안속성들을 바란다면 보안모형의 복잡성을 제한하는것이 좋을것이다. 그렇게 하면 그 모형이 요구하는 모든 보안속성들을 묘사할수 없을수도 있지만 《보안》을 증명하는 효과적인 방법을 얻을수는 있다 . 다시말하여 간단한 모형으로 충분히 묘사될수 있는 단순한 체계를 설계하는것이 좋다. 체계와 모형사이에 지나치게 넓은 간격이 있으면 그 모형에서의 보안의 증명을 담보하기는 힘들다.

보안속성들과 서술가능한 체계의 두가지 측면에서 볼 때 보다 표현적인 보안모형일수록 보안속성들을 증명하기가 더 어렵다.

제4절. 《장성》모형

브레워(Brewer)와 나쉬(Nash)에 의해 제안된 《장성》(Chinese wall)모형은 상담업무에서 분석자가 서로 다른 의뢰자들(회사들)을 취급할 때 리해관계의 충돌이 일어 나지 않도록 접근규칙들을 모형화한다[23]. 흔히 충돌은 의뢰자들이 같은 시장에서 직접적인 경쟁자들이거나 또는 회사들의 소유권문제로 인하여 일어 난다.

분석자는 다음의 보안방책을 준수하여야 한다.

규칙: 리해관계의 충돌을 일으키는 정보흐름이 없어야 한다.

이 방책을 처리하려면 벨-라파둘라모형의 상태모임을 이 방략에 맞도록 일부 개작해야 한다.

- 회사들의 모임을 C로 표시한다.
- 객체들의 모임 O는 단일회사와 관련된 정보항목이다. 분명히 분석자는 주동체이며 S는 주동체들의 모임이다.

- 같은 회사와 관련된 모든 객체들은 회사자료모임에 수집된다. 함수 $y:O\rightarrow C$는 매 객체의 회사자료모임을 준다.
- 리해관계클라스들의 충돌은 회사들이 경쟁관계에 있다는것을 나타낸다. $x:O\rightarrow P(C)$ 는 매 객체에 대하여 리해관계클라스의 충돌을 준다. 즉 객체의 내용들에 대하여 학습하지 말아야 할 모든 회사들의 모임을 준다.
- 객체 o의 보안표식은 $(x(o), y(o))$쌍이다.
- 소독된(sanitised)정보는 민감한 상세들을 없앴으므로 접근제한을 받지 않는다. 소독된 객체 o에 대하여 $x(o)=\phi$를 설정한다.

리해관계의 충돌은 최근에 접근된 객체들뿐아니라 과거에 접근된 객체들로부터도 일어 난다. 그러므로 주동체들의 행위리력을 기록하는 수단이 필요하다.이를 위하여 론리적인(Boolean) $S\times O$행렬 N을 도입한다. 여기서

$$N_{s,o}=\begin{cases}\text{참: 주동체 } s\text{가 객체 } o\text{에 접근하였다면}\\ \text{거짓: 주동체 } s\text{가 객체 } o\text{에 접근하지 않았다면}\end{cases}$$

만일 모든 $s\in S$와 모든 $o\in O$에 대하여 $N_{s,o}$=거짓(false)으로 설정하면 안전한 초기상태로 된다.

첫번째 보안방책은 직접적인 정보흐름을 취급한다. 여기서는 주동체가 리해관계의 충돌에 로출되는것을 막아야 한다. 그러므로 접근은 요구된 객체가 다음에 속할 때에만 허락된다.

- 이미 사용자가 가지고 있는 회사자료모임
- 리해관계클라스의 완전히 서로 다른 충돌

형식적으로 이 ss-속성을 다음과 같이 표현할수 있다.

ss-속성: $N_{s,o'}$ = 참(true)을 가지는 모든 객체 o'에 대하여 $y(o)\notin x(o')$이거나 $y(o)=y(o')$일 때에만 주동체 s는 객체 o에 접근할수 있다.

이 속성은 스스로는 서술된 보안방책을 충분히 실현할수 없다. 직접정보흐름은 여전히 가능하다. 다음실례를 보자. 두 경쟁자인 회사_A와 회사_B는 같은 은행에 자기들의 구좌(등록자리)를 가지고 있다. 분석자_A는 회사_A와 은행업무를 취급하면서 회사_A에 대한 중요한 정보로써 은행문서를 갱신한다. 한편 회사_B와 은행업무를 취급하는 분석자_B가 경쟁자의 문서에 대한 정보에 접근한다고 하자. 이때 쓰기접근을 통제하기 위해 *-속성을 도입한다.

***-속성**: 주동체 s가 $y(o)=y(o')$이고 $x(o')\neq\phi$인 객체 o'에 대한 읽기접근을 하지 않을 때에만 주동체 s에게 객체 o에 대한 쓰기접근이 허락된다.

객체에 대한 쓰기접근은 어떤 다른 객체도 읽을수 없을 때에만 허락된다(그 객체들은 서로 다른 회사자료모임의 그리고 비공개정보를 포함하는 객체들이다). 접근권한들이 항상 정적이라고 가정하는 BLP와 대조적으로 매 상태이행에서 접근권한들을 다시 검토해야 하는 모형을 보기로 한다.

제5절. 비바모형

비바(Biba)모형 [19] 은 BLP와 매우 류사한 상태기계모형을 리용하여 주동체의 객체들에 대한 접근의 의미에서 완정성을 취급한다. 완정성준위의 살창 (L, $\leq$)이 있다. 함수 $fs:S \to L$과 $fo:O \to L$은 주동체들과 객체들에 완정성준위를 할당한다. 이 준위들은 완정성방책을 표현하는 기초를 이룬다. 여기서는 높은 준위실체를 《깨끗한》것으로, 낮은 준위실체를 《오염된》것으로 대응시킨다. 완정성살창에서 정보는 오직 아래방향으로만 흐를수 있다. BLP와 달리 단일한 높은 준위완정성방책은 없다.

대신에 변화된 방법들을 보게 된다. 일부는 지어 서로 모순되는 방책들을 만들어 내기도 한다.

1. 정적완정성준위

BLP의 완정속성을 본따서 완정성준위가 변하지 않는 방책들을 서술할수 있다. 다음의 두가지 완정성속성들은 2중의 위임된 BLP방책들이다.

- 단순완정성속성: 만일 주동체 s가 객체 o를 수정(변경)할수 있다면 $fs(s) \leq fo(o)$이다 (No write-up).
- 완정성 *-속성 : 만일 주동체 s가 객체 o를 읽을수(관찰) 있다면 s는 $fo(p) \leq fo(o)$인 때에만 일부 다른 객체 p에 대한 쓰기권한을 가질수 있다.

이 두가지 방책들은 깨끗한 주동체들과 객체들이 오염된 정보에 의해 오염되는것을 막는다.

2. 동적완정성준위

《장성》모형과 류사하게 다음의 두가지 완정성속성들은 어떤 실체가 낮은준위정보와 접촉하였다면 그의 완정성준위를 자동적으로 조정한다.

- 주동체낮은내비침무늬속성(Subject Low watermark property): 주동체 s는 임의의 완정성준위에서 어떤 객체 o를 읽기(관찰)할수 있다. 주동체의 새로운 완정성준위는 $\inf(fs(s), fo(o))$ 이다. 여기서 $fs(s)$와 $fo(o)$는 조작전의 완정성준위들이다.
- 객체낮은내비침무늬속성(Object low watermark property): 주동체 s는 임의의 완정성준위에서 어떤 객체 o를 수정(변경)할수 있다. 그 객체의 새로운 완정성준위는 $\inf(fs(s),fo(o))$이다. 여기서 $fs(s)$와 $fo(o)$는 조작전의 완정성준위들이다.

$fs(s)$와 $fo(o)$의 가장 큰 아래한계인 완정성준위 $\inf(fs(s), fo(o))$는 여기서 완정성준위살창을 취급하므로 잘 정의된다.

3. 호출을 위한 방책

비바모형은 접근조작인 호출(invoke)을 포함하도록 확장할수 있다. 어떤 한 주동체가 다른 주동체 례하면 쏘프트웨어도구를 어떤 객체에 접근하도록 호출할수 있다. 이것은 중간층준위들에서 접근조종을 정식화하는 방향에로의 첫 걸음이다. 어떤 종류의 방책들이 호출을 통제하여야 하는가? 호출이 위임완정성방책들을 우회하지 못한다는것을 확인하기를 원하는가? 다음속성을 추가할수 있다.

- 호출속성: $fs(s_2) \le fs(s_1)$인 때에만 주동체 s_1 는 주동체 s_2을 호출할수 있다.

주동체들은 보다 낮은 준위에 있는 도구 (tool)들만을 호출하는것이 허락된다. 한편 오염된 주동체가 깨끗한 객체에 접근하기 위해 깨끗한 도구를 리용할수 있고 깨끗한 객체를 오염시킬수 있다.

또한 이를 위해 틀림없이 도구들을 리용하려 할수 있다. 오염된 주동체들이 깨끗한 어떤 객체에 접근해야 하는데 그러자면 오직 깨끗한 도구를 리용하여야만 한다. 이 도구는 그 객체가 깨끗한가를 확인하기 위해 수많은 일치성검사를 수행할수 있다. 이 경우에 깨끗한 주동체가 오염된 도구들을 리용하는것을 바라지 않을것이며 다음속성을 채용할수 있다.

- 고리속성: 주동체 s_1는 모든 완정성준위들에서 객체들을 읽을수 있다. 그것은 다만 $fo(o) \le fs(s)$인 객체 o들을 수정할수 있으며 $fs(s_1) \le fs(s_2)$인 때에만 주동체 s_2을 호출할수 있다.

분명히 마지막 두개 속성들은 서로 모순되며 보다 적절한 결심을 하기 위해서는 응용을 보아야 한다.

제6절. 클라크-윌슨모형

클라크(Clark)와 윌슨(Wilson)은 상업적응용에서의 보안요구들을 취급한다[32]. 그들은 이 요구들이 주로 자료완정성 례하면 자료의 권한이 없는 수정, 협잡, 오유 등을 막기 위한것이라고 주장한다. 이것은 완정성에 대한 보다 넓은 정의이다. 실제로 그들은 보안의 범위밖인 동시성조종문제들까지 취급한다. 완정성요구들은 다음의 두개 부분으로 나누어 진다.

- 내부일관성: 체계의 내부상태의 속성에 기인하며 계산체계에 의하여 실행될수 있다.
- 외부일관성: 실세계에 대한 체계의 내부상태의 관계에 기인하며 계산체계외부의 수단 례하면 검열(auditing)에 의하여 시행되여야 한다.

완정성을 요구하는 일반적기구들은 다음과 같다.

- 잘 형성된 거래(well-formed transaction): 자료항목들은 특정한 프로그람들의 모임에 의해서만 취급될수 있다. 사용자들은 자료항목들보다도 프로그람들에 접근한다.
- 임무의 분담: 사용자들은 자료를 취급하기 위하여 협력하여야 하며 보안체계를 파악하기 위해 협의하여야 한다.

임무의 분담은 안전체계의 조작에서 가끔 나타난다. 서로 다른 사람들이 체계를 개발, 시험, 인증을 하고 조작할것을 요구하는것은 응당한것이다. 다시말하여 조작과정에 사람들은 서로 거래를 위하여 협력해야 한다.

클라크-윌슨 모형은 프로그람을 주동체와 객체(자료항목)사이의 중간조종층으로 리용한다. 주동체들은 정해 진 프로그람들을 실행할 권한을 가진다. 한편 특정한 프로그람들을 통해서만 자료항목들을 호출할수 있다. 추상자료형[39]과 객체지향프로그람작성에서처럼 특정한 형태의 자료에 접근할수 있는 프로그람들의 모임을 정의하는것은 쏘프트웨어공학에서 일반적인 수법이다. 이것이 안전체계를 구성하는데서는 필수적인것으로 리용될수 있다. 클라크와 윌슨이 쓴 《보안준위대신에 프로그람으로써 주동체와 객체들을 분류하기》는 BLP의 효과에 대한 증언이다. 클라크-윌슨 모형에서 완정성은

어떤 자료항목에 접근할수 있는 프로그람을 실행할 권한을 가지는것

을 의미한다. 클라크와 윌슨은 군사적 및 상업적보안요구들사이 차이를 강조하고 있다. 이 두 세계에서 기밀성과 완정성의 상대적중요성이 동일한것은 아니다. 군사적응용에서는 완정성요구를 가진다면 상업적응용에서는 기밀성요구를 가질것이다. 이 책의 목적에서 볼 때는 더 많은 상대적차이가 있다. 클라크-윌슨모형에서 접근조작들은 복잡한 응용지향의 처리들을 수행하는 프로그람들이다. BLP에서의 접근조작들은 조작체계에 적합한 단순하고 일반적인것들이다. 여기서는 일반용조작체계(BLP)와 응용지향IT체계(클라크-윌슨)사이의 차이를 보기로 한다. 총체적으로 클라크-윌슨모형에서는 다음과 같은 점들을 고려한다.

1. 주동체들은 식별되여야 하며 권한을 가져야 한다.
2. 객체들은 제한된 프로그람들의 모임에 의해서만 조작될수 있다.
3. 주동체들은 제한된 프로그람들이 모임만을 실행할수 있다.
4. 적절한 검열기록이 보존되여야 한다.
5. 체계가 원만히 작업한다는것을 확인하여야 한다.

이 모형의 형식화에서 보안방책에 의해 통제되는 자료항목들을 속박된 자료항목(CDIS)이라고 한다. 체계에로의 입력들은 속박되지 않는 자료항목들(UDIS)로서 취해진다. UDIS의 CDIS에로의 변환은 체계에 있는 보안기구가 단독으로 조종할수 없는 체계의 림계부분이다. CDIS는 오직 변환수속들(TPS)에 의해서만 취급될수 있다. 상태의 완정성은 완정성확인수속들(IVS)에 의해서 검사된다.

보안속성들은 다섯가지 확인(certification)규칙들을 통하여 정의된 한 측면이며 보안방책이 응용요구들에 부합되게 하는 검사들을 제안한다.

1. IVP들은 그 IVP가 실행될 때 모든 CDI들이 유효한 상태에 있다는것을 담보하여야 한다.
2. TP들이 유효하다는것을 확인하여야 한다. 즉 유효한 CDIs들은 유효한 CDI들로 변환되여야 한다. 매개 TP는 특정한 CDI들의 모임에 접근하기 위해 확인된다.
3. 접근규칙들은 임무요구들의 임의의 분리를 만족시켜야 한다.
4. 모든 TP들은 추가만이 가능한 기록에 써야 한다.
5. 입력으로서 UID를 가지는 임의의 TP는 UID를 CDI로 변환하거나 또는 UDI를 거부하고 어떤 변환도 수행하지 않아야 한다.

다음의 4가지 시행규칙들은 보안방책을 실시하여야 할 콤퓨터체계내의 보안기구들을 서술한다. 이 규칙들은 BLP에서의 자유접근조종과 일부 류사성을 가진다.

1. 체계는 TP가 접근하도록 확인되는 CDI들을 주는 기입항목들(TPi: CDIa, CDIb, …)의 목록을 보존하고 보호하여야 한다.
2. 체계는 사용자들이 실행할수 있는 TP 들을 명시하는 기입항목들(UserID, TPi: CDIa, VDIb, …)의 목록을 보존하고 보호하여야 한다.
3. 체계는 TP를 실행할것을 요구하는 매개 사용자를 인증하여야 한다.
4. 어떤 TP를 위한 접근규칙을 확인할수 있는 주동체만이 목록에서 해당한 기입항목을 수정할수 있다. 이 주동체는 그 TP에 대한 실행권한을 가지지 말아야 한다.

끝으로 클라크-윌슨모형은 어떤 특정한 보안방책의 모형이라기보다 보안방책들을 위한 골격과 안내(《모형》)이라는것을 강조한다.

제7절. 정보흐름모형

벨-라파둘라의 모형에서 정보는 잠복통로를 통하여 높은 보안준위로부터 낮은 보안준위에로 흐를수 있다. 정보흐름모형은 BLP에 의해 모형화된 접근조작들을 통한 직접정보흐름뿐아니라 임의의 종류의 정보흐름을 모형화한다. 만일 y를 고찰함으로써 x에 대하여 더 알수 있다면 상태이행은 객체 x로부터 객체 y에로의 정보흐름을 일으킨다. 이미 x에 대해 알고 있다면 x로부터 아무런 정보도 흐를수 없다. 이로부터 다음과 같이 구분할수 있다.

- 명시적인 정보흐름: 할당 $y := x$ 후에 y를 관찰하면 x의 값을 알수 있다.
- 암시적인 정보흐름: 조건문 if x=0 then y:=1 후에 y를 관찰하면 할당 y:=1이 실행되지 않았다 해도 x에 대해서 무엇인가 알수 있다. 실례로 만일 y=2이면 $x \neq 0$이라는것을 알수 있다.

정보흐름에 대한 엄밀하고 정량적인 정의는 정보리론에서 준다. x로부터 y에로의 정보흐름은 주어 진 y의 값에 대한 x의 애매성(조건적엔트로피)의 변화에 의해서 측정된다. 정보흐름모형의 요소들은 다음과 같다.

- 보안표식들의 살창(L, $\leq$),
- 표식 붙은 객체들의 모임,
- 보안방책: 표식 c_1를 가진 객체로부터 표식 c_2을 가진 객체에로의 정보흐름은 $c_1 \leq c_2$인 때에만 허락된다. 이 규칙에 위반되는 어떤 정보흐름도 금지한다.

체계는 위법인 정보흐름이 없을 때 안전하다고 한다. 이러한 모형의 우점은 그것이 모든 종류의 정보흐름을 포괄한다는것이다. 결함은 안전체계를 설계하기가 보다 어려워진다는것이다. 주어 진 체계가 정보흐름모형에서 안전한가 아닌가를 검사하는것은 비결정성문제라는것이 알려 졌다.

나아가서 정보흐름방책들의 정적 및 동적시행을 구분하여야 한다. 첫째 경우에 체계(프로그람)는 정적객체로 고찰된다. 둘째 경우는 실행중의 체계를 고찰한다. 이로부터 우리는 일부 정보흐름이 리론적으로는 가능할수 있으나(따라서 정적해석에서 검출되여야 한다.) 실행시에는 절대로 일어 나지 않을것이라는것을 알수 있다. 그러므로 정적해석은 너무 제한적인 체계들을 만들어 낼수 있다.

비간섭모형(Non-interference model)들은 정보흐름모형들을 대신할수 있다.그것들은 주동체가 체계의 상태에 대하여 알고 있는것을 서술하기 위한 다른 수학적표현을 준다. 주동체 s_1의 작용이 s_2이 체계를 보는데 아무런 영향을 주지 않으면 주동체 s_1은 주동체 s_2과 간섭하지 않는다. 현재 정보흐름모형과 비간섭모형은 연구단계에 있으며 안전체계의 설계를 위한 실천적방법론의 기초이다.

이 장의 문헌안내

보안모형들은 흔히 보안방책들의 형식화로 간주된다. 이 정의는 보안방책으로 무엇을 하려 하는가를 안다고 가정한다. 보안방책은 기업의 보안요구들을 받아 들이거나 또는 보안을 구축하기 위한 단계들을 서술할수 있다. 보안방책이라는 용어의 여러가지 의미에 대한 론의는 [147]에 주어 진다. 상업적조직들에서의 보안방책을 정의하는 현수준은 [143]에 서술되며 위험분석의 실천적측면들은 [31]에 주어 진다.

보안모형들에 대한 10년간의 기사들을 따로 묶어서 [84]와 [98]에 준다. 벨-라파둘라모형에 대한 초기론문들이 최근 Journal of computer security [12]에 재출판되였다. 벨-라파둘라모형에서 접근권한을 변경시키는 방책들의 골격은 [97]에서 론의된다. 연구문헌들은 전부BLP의 MLS 방책들을 여전히 시행하면서 BLP모형의 범위를 넓히는 기고작품들이다. 클라크와 윌슨에 의한 초기론문은 많이 요구되는 문헌이다[32]. 자격을 리용하는 클라크-윌슨모형의 실현은 [75]에 서술된다. 클라크-윌슨모형을 실현하기 위해 리용할수 있는 위임의 완정성조종을 제공하는 비바모형의 확장이

[87]에 주어 진다. HRU모형과 정보흐름모형의 결정가능성속성들의 상세한 정의, 증명, 정리들과 함께 [39]에 주어 진다. 비간섭성모형들에 대해서는 문헌[58]을 찾아 보시오 .

련습문제

1. 기본접근방식 변경(alter)과 관찰(observe)을 리용하여 *-속성을 서술하시오.

2. 벨과 맥클린의 론쟁에 대한 자신의 의견을 소론문으로 쓰시오.

3. 벨-라파둘라는 접근권한을 변경하는 방책을 표현하지 못한다.
   당신은 어떤 방책을 제안하는가?

4. 《장성》모형을 벨-라파둘라공격에 맞출수 있는가?

5. 《장성》모형에서 *-속성은 현재 읽기접근에만 의거하는가 아니면 임의의 과거의 읽기접근에 의거해야 하는가?

6. 비바모형은 여러가지 완정성방책들을 포함한다. 다음것들이 합당한 응용령역들의 실례를 드시오.

   - 정적완정성표식을 가지는 방책,
   - 동적으로 변하는 완정성표식을 가지는 방책,
   - 고리속성.

7. 벨-라파둘라와 비바를 기밀성과 완정성을 동시에 모형화하는데 리용할수 있는가?

8. 클라크-윌슨시행규칙들을 서술하는 형식적모형을 만드시오.

9. 30년후에 해제될 문서들을 위한 보안모형을 개발하시오.

10. 환자기록들과 처방들에 대한 접근을 조종하는 의학정보체계에서

    - 의사들은 환자기록들과 처방들을 읽거나 쓸수 있다.
    - 간호원들은 처방들을 읽거나 쓸수 있지만 환자기록의 내용에 대해서는 아무것도 학습하지 말아야 한다.

환자기록으로부터 처방에로의 정보흐름을 막는 살창모형에 이 방책을 어떻게 적용할수 있는가? 당신의 의견에는 어느 보안모형이 이 방책에 가장 적합한가?

제5장. 보안핵심

앞에서 본 두개의 장에서는 접근조종방책들을 세우기 위한 여러가지 방안들을 소개하고 다음에 그 응용을 연구하며 가장 적절한 접근조종구조들과 보안모형들을 결정하였다. 선택한 보호기구들을 어떻게 실현하겠는가? 앞에서 본 계층화된 체계모형의 구조내에서 다음의 두가지 질문에 대답하여야 한다.

- 접근조종은 어디에 위치하는가?
- 선택한 방안이 추가적인 보호요구들을 고려하게 되여 있는가？

이 장은 앞에서 본 모형에서 맨 아래 두개 층들에서의 보안기구들을 고찰한다.

목적

- 낮은 체계층에서 보안을 실시하는 리유를 설명한다.
- 낮은 체계층들에서 쓸수 있는 보안기구들의 륜곽을 얻는다.
- 두개의 중요한 보안원시조작들로서 상태들과 조종된 호출(Invocation)을 도입한다.
- 보안모형들이 보안핵심을 해석하는데 어떻게 리용될수 있는가를 본다.

제1절. 리론적기초

낮은 체계층들중의 하나에 보안을 설치하는데는 두가지 리유가 있다(그림 5-1). 임의의 주어 진 층에 있는 보안기구는 공격자가 그보다 낮은 층에 있으면 효과가 없다. 따라서 체계의 보안을 평가하자면 자기의 보안기구를 우회할수 없다는것을 담보하여야 한다. 체계가 복잡할수록 이 검사는 더 어려워 진다. 체계의 핵심부에서 철저한 해석을 할수 있는 적절하고도 단순한 구조를 찾아야 한다. 이 론거가 핵심부에 보안을 배치하는 첫번째 리유이다.

높은 준위의 담보를 위해 보안을 평가할수 있다.

극소형처리기설계는 대부분의 사용자들에게 가장 유용한 조작들의 모임을 제정하는 매우 중요한 과학이다. 일반조작들의 옳은 선택과 효과적인 실현은 전체적인 성능을 결정한다. 보안을 실현하는데서도 같은 방법을 택할수 있다. 일반보안기구들을 결정하고 그것들을 체계의 핵심부에 넣는다. 이것이 핵심부에 보안을 설치하는 두번째 리유이다.
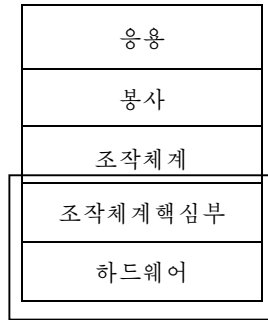
그림 5-1. IT체계의 층

그림 5-2. 사람-기계척도에서 보안핵심의 위치

보안기구를 체계의 핵심부에 넣는것은 보안으로 인한 간접소비를 감소시킨다.

이 장은 Motorola 68000과 Intel 80386/486을 실례로 하여 극소형처리기들의 보안특징들을 보고 Multics조작체계의 참조감시기를 고찰한다. 그것은 사람-기계척도에서 기계쪽에 치우쳐 배치된 보안기구들을 포함한다(그림 5-2). 이 책의 뒤부분에서 매 층들을 따라 가면서 조작체계들에 의해서 제공된 보안기구들(제 6 장과 제 7 장)과 자료기지관리체계(제14장)와 같은 쏘프트웨어봉사들(middleware)에 의해 제공된 보안기구들의 실례들을 제시한다.

제2절. 조작체계의 완정성

콤퓨터보안에는 세가지 기본개념이 있는데 그것들은 서로 밀접히 련관되여 있어 혼돈하기 쉽다. 그러나 따로따로 분리할수는 있다. 여기서는 오렌지부크[112]어휘집을 참고로 하여 정의한다.

- **참조감시기**(Reference Monitor): 주동체의 객체에로의 모든 접근들을 중개하는 추상기계를 가리키는 접근조종개념
- **핵심부에서의 보안**: 참조감시기개념을 실현하는 신용계산기지인 하드웨어, 펌웨어, 쏘프트웨어요소들, 모든 접근들을 중개하여야 하며 수정을 가할수 없고 정확성을 담보할수 있어야 한다.
- **신용계산기지**(Trusted computing base)(TCB): 콤퓨터체계내의 보호기구들의 총체(하드웨어, 펌웨어, 쏘프트웨어를 포함하는) 즉 이것들의 조합은 보안방책실현을 책임진다. TCB는 어떤 제품이나 체계우에서 단일한 보안방책을 함께 시행하는 하나 또는 그이상의 요소들로 구성된다. 보안방책을 정확히 수행하기 위한 TCB의 능력은 다만 그 TCB내에 있는 기구들과 보안방책과 관련되는 파라메터(례하면 사용자의 기밀취급허가)들에 대한 체계관리성원들의 정확한 입력에만 의존한다.

참조감시기는 하나의 추상적인 개념이며 보안핵심은 그의 실현이다. TCB는 다른 보안기구들중에서 보안핵심을 포함한다. 그림 5-1의 두개의 바닥층에 있는 보안기구들은 대략적으로 보안핵심에 대응된다. 조작체계의 보안핵심을 될수록 단순하게 하는것은 안전한 조작체계를 설계하고 평가하는데서 본질적의의를 가진다. 일부 책들에서 TCB들을 BLP류사방책을 시행하는 보안핵심들과 거의 동의어로 취급하고 있는데 정의로부터 알수 있는바와 같이 이것은 오렌지부크저자의 의도와는 맞지 않는다.

보안기구들을 체계의 핵심부에 넣는것을 지지하는 방향으로 진행된 모든 론의들은 우리를 사람-기계척도에서 기계쪽끝으로 치우치게 하였다. 그 결과 다음과 같은것을 예측할수 있다.

보안핵심에 의하여 만들어 진 접근조종결심은 응용에 의하여 만들어 진 접근조종결심과 많이 차이난다.

이제 보안핵심에 포함될 보안기구들을 일반용어로 나타내 보자. 모든 접근조종방책들을 시행할수 있는 조작체계를 가지고 있다고 가정하자. 조작체계가 예견대로 동작하는 한 자원들에 대한 권한이 없는 접근은 불가능하다. 물론 이것은 바로 공격자에게는 단서로 된다. 공격자는 보호기구들을 우회하기 위해 조작체계를 수정하여 보안조종이 불가능하게 하려고 할수 있다. 본래는 기밀성과 관련된 문제이지만 완정성문제에 직면하게 된다. 조작체계는 접근요구들의 조정자일뿐아니라 그자체가 접근조종의 대상이다. 새로운 보안방책은 다음과 같다.

규칙 : 사용자들은 조작체계를 수정할수 없어야 한다.

이것은 응당 강력하고 효과적인 지원을 받아야 할 일반적인 보안방책이다. 체계를 안전하게 하기 위해 다음의 두가지 상반되는 요구를 처리하여야 한다.

- 사용자는 조작체계를 리용(호출)할수 있어야 한다.
- 사용자는 조작체계를 잘못 리용하지 말아야 한다.

이 목적들을 달성하기 위하여 일반적으로 리용되는 두가지 중요한 개념들은 상태정보와 조종된 호출(또한 제한된 특권이라고도 한다.)이다. 이 개념들은 응용쏘프트웨어, 조작체계, 하드웨어 등 체계의 어느 층에서도 리용될수 있다. 그러나 공격자가 보다 낮은 층에로의 접근을 얻는다면 이 기구들은 무능력하게 된다.

1. 조작방식들

사용자들로부터 조작체계자체를 보호하기 위한 첫째 전제조건은 사용자를 대표하여 한 계산과 조작체계를 대표하여 한 계산(computation)을 구별하는 능력이다. 이러한 목적으로부터 조작체계가 여러가지 방식에서 작업할수 있게 하는 상태기발을 리용한다. 다음의 두가지 실례는 처리기준위에서 이것을 례증한다.

- Motorola 68000: 한개 상태비트로 사용자방식과 감시자(Supervisor)체계방식을 구분할수 있게 한다.
- Intel 80386: 두개의 상태비트로 4가지 방식을 구분할수 있게 한다.

Unix조작체계는 감시자(뿌리)방식과 사용자방식으로 구분된다.

그러면 왜 이러한 방식들이 필요하겠는가? 실례로 사용자가 기억기에 직접 쓰거나 론리파일구조를 다치는것을 막기 위해 조작체계는 처리기가 관리자방식에 있을 때에만 기억위치에 대한 쓰기접근을 허락할수 있다.

2. 통제된 호출

앞에서 본 실례를 계속한다. 사용자는 감시자방식을 요구하는 조작 례하면 기억위치에 대한 쓰기를 하려고 한다. 이 요구를 처리하기 위해 처리기는 방식을 절환하여야 하는데 이 절환이 어떻게 수행되여야 하겠는가? 간단히 상태비트를 감시자방식으로 변경시키는것은 사용자가 실제로 무엇을 하는가에 대한 어떤 조종도 없이 사용자에게 이 방식과 관련되는 모든 특권을 주어 버리는것으로 될것이다. 그러므로 체계는 감시자방식의 미리 정의된 조작모임을 수행하고 사용자방식으로 되돌린다. 그후에야 조종처리는 사용자에게 되돌아 온다. 이러한 처리를 조종된 호출이라고 한다.

제3절. 하드웨어보안특징

하드웨어는 IT체계에 대한 보안모형에서 가장 낮은 층이다. 하드웨어는 콤퓨터보안이 물리적보안과 련결될수 있는 곳이기도 하다. 따라서 하드웨어준위에 있는 보안기구들은 우리의 연구에서 출발점이라고 볼수 있다. 여기서는 독자들이 콤퓨터구성방식의 기본개념들을 잘 알고 있으리라고 가정한다.

1. 콤퓨터구성방식의 간단한 개괄

그림 5-3에 중앙처리장치(CPU), 기억기 그리고 CPU와 기억기를 련결하는 모선으로 구성된 콤퓨터의 간단한 도해를 주었다. 실제로 세개의 실체들은 모두 훨씬 세부적인 구조를 가질수 있다.
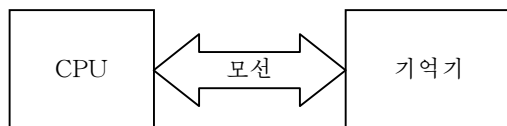
그림 5-3. 콤퓨터의 도해

중앙처리장치

CPU의 기본요소들은 다음과 같다.

- **산수론리단**(ALU): 기계언어로 주어 진 명령들을 실행한다.명령의 실행은 상태등록기에 있는 비트들을 설정할수도 있다.
- **등록기들** : 일반용등록기들과 전용등록기들이 있다. 중요한 전용등록기들은 다음과 같다.
- **프로그람계수기** : 실행되여야 할 다음명령을 포함하는 기억위치를 지적한다.
- **탄창지시기** : 체계탄창의 꼭대기를 가리킨다.
- **상태등록기**: CPU가 상태정보를 보관하도록 한다.

체계탄창은 특별히 지정된 기억기의 부분이다. 탄창은 그의 꼭대기에로 자료를 밀어넣거나 그 꼭대기로부터 자료를 꺼내는 식으로 접근될수 있다. 서로 다른 처리들사이를 절환하기 위해 CPU는 상태절환을 실행하며 현재처리의 상태 즉 프로그람계수기, 상태등록기 등을 새로운 처리에로 조종을 넘겨 주기전에 탄창에 보관한다.

추상의 위험에 대하여 경고한적이 있는데 탄창은 그러한 개소의 하나이다. 추상적인 탄창의 크기가 무한하다는것은 옳은 주장이다. 그러나 실현에서 는 고정된 기억구역을 탄창을 위해 할당할수 있다. 만일 탄창이 그의 최대크기를 벗어 나면 보안문제들이 꼭 생길것이다.

기억구조

다음의 간단한 개괄에서 여러 기억구조들의 보안특징들에 특별히 주목해야 한다.

- RAM(random access memory): 이것은 읽기쓰기기억기이다. 여기서 완정성과 기밀성의 문제들을 고려해야 한다.
- ROM(read-only memory): 여기서는 기밀성의 문제만을 고려하면 된다. ROM은 조작체계를 기억시키는 위치이다.

- EPROM(erasable and programmable ROM): 조작체계의 부분이나 암호열쇠들을 기억시켜 두는데 리용할수 있다. 기술적으로 보다 세련된 공격들은 이 부분을 위협할수 있다.
- WROM(write-once memory): 기억기구조는 기억내용을 한번만 고착시킬수 있게 되여 있다. 하드웨어에서 이것을 쓰기선에 배치된 휴즈를 끊어 버림으로써 실현할수 있으나 론리적휴즈를 생각할수도 있다. WROM은 암호열쇠를 보관하는데 알맞는 위치이다. 쓰기만하는 디스크들은 검열추적들을 기록하는데 리용된다.

휘발성기억기와 불휘발성(영구)기억기라는 구분도 있다. 휘발성기억기는 전원이 꺼지면 그의 내용이 없어 진다. 물리적으로 이 공정은 즉시에 일어 나는것이 아니며 완전히 없어 지는것도 아니다. 만일 전원이 꺼진후 곧 다시 켜면 낡은 자료가 기억기에 여전히 남아 있을수 있다. 지어는 전원이 일정한 시간동안 꺼져 있었어도 낡은 기억내용들이 특별한 전자기술에 의해 재생될수 있다. 때문에 기억기에 그 기억매질에 의존하는 적당한 비트모양으로 반복적으로 겹쳐쓰기하여야 한다[113].

영구기억기는 전원이 꺼져도 그의 내용을 보존한다. 만일 암호화열쇠와 같은 중요자료들이 영구기억기에 기억되여 있고 공격자들이 CPU를 우회하여 기억기에 직접 접근하려고 한다면 암호화적인 또는 물리적인 보호와 같은 추가적인 기구 등이 실현되여야 한다. 실례로 빛수감기는 손대지 못하도록 모듈안에 설치되여 있고 손조작의 시도를 검출하여 모듈안에 보존된 자료의 삭제를 유발시키도록 되여 있다. 물리적보호에 대하여서는 이 책에서 설명하지 않는다. 우리는 사용자가 CPU를 통해서만 기억기에로의 접근을 실현할수 있는 경우에로 범위를 제한하고 CPU가 어떻게 기밀성과 완정성을 실현할수 있겠는가를 연구한다. 실례로 콤퓨터비루스가 오염된 판본으로 조작체계의 깨끗한 판본을 파괴하는것을 예방하기 위해서는 어떻게 하여야 하겠는가?

그림 5-3의 《기억기》 역시 하나의 추상이라는것을 상기하는것이 중요하다. 론리적으로 기억기는 주기억기, 빠른 접근을 위한 완충기억기, 완충기 등으로 구성될수 있다. 여벌기억매체도 이 목록에 포함될수 있다. 따라서 하나의 자료객체가 이러한 기억기계층에서 하나이상의 위치에 동시에 존재할수 있다. 기억기에서의 영구기록외에 림시기록이 있다. 보통 이러한 림시기록의 위치와 수명은 사용자가 조종할수 없다. 그러나 림시기록들중의 하나가 보호되지 않은 기억령역에 있다면 사용자는 자료객체에 대한 보안조종을 우회할수 있다.

2. 처리와 스레드

처리란 실행중에 있는 어떤 프로그람이다. 그러므로 처리는 조작체계는 물론 보안을 위한 중요한 조종의 단위이다. 풀어서 말하면 처리는 다음과 같은것들로 구성된다.

- 실행가능한 코드
- 자료
- 실행환경 례하면 일정한 관계되는 CPU등록기들의 내용들

처리는 자기의 전용주소공간에서 작업하며 다른 처리들과는 오직 조작체계에 의해서 제공되는 기본지령(PRIMITIVE)들을 통해서만 통신할수 있다. 처리들사이의 이러한 론리적구분은 보안을 위한 유용한 기초이다. 한편 처리들사이의 상태절환은 조작체계가 전체 실행상황을 탄창에 보관하여야 하는 비용이 많이 드는 조작이다.

토막과제(스레드)란 하나의 처리내에 있는 실행의 가지들이다. 토막과제들은 하나의 주소공간을 공유함으로써 완전한 상태절환의 쓸모 없는 시간소비를 피하며 한편 보안기구에 의한 조종을 피할수 있다.

3. 통제된 호출 -새치기

처리기들은 프로그람이나 사용자요구, 하드웨어고장 등의 오유들에 의해서 발생된 실행의 새치기을 처리하도록 장비된다. 그러한 기구들을 례외, 새치기, 중단이라는 여러가지 이름으로 부른다. 서로 다른 용어로서 서로 다른 형태의 사건들에 대응될수 있지만 사실은 류사한 분류들이다. 실례로 [66]을 참고하시오. 구체적인것은 제5장 6절을 참고하시오.

여기서는 일반용어로 《중단》을 사용하며 중단들이 어떻게 보안목적에 리용되는가를 설명한다. 중단은 새치기벡토르표에 있는 하나의 주소를 포함하는 CPU에 입력하는 새치기벡토르입력이다. 새치기표는 중단에 명시된 사건을 처리하는 프로그람의 위치를 준다. 이 프로그람을 새치기처리기라고 한다. 중단이 일어 나면 체계는 그의 현재상태를 탄창에 보관하고 다음 새치기처리기를 실행한다(그림 5-4). 이런 방법으로 조종은 사용자프로그람으로부터 떨어 져 나온다. 처리기는 사용자프로그람에 조종을 되돌려 주기전에 관리자상태비트를 지움으로써 체계가 본래의 상태로 복귀된다는것을 확인해야 한다.
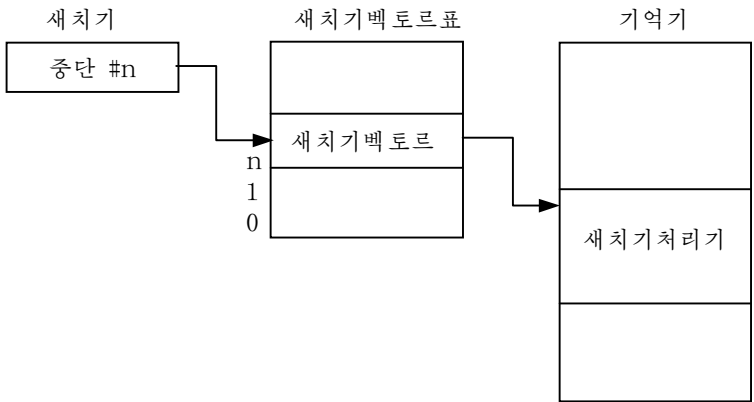
그림 5-4. 새치기의 처리

처리기가 현재새치기를 처리하는 동안에 또 다른 새치기가 일어 날수도 있다. 그때 처리기는 현재의 새치기처리기를 새치기하여야 한다. 이러한 상황을 잘못 조종하면 보안류출을 초래할수 있다. 실례로 사용자가 CTRL-C를 누름으로써 프로그람의 실행을 새치기시키고 처리기가 현재처리의 상태비트를 가지고 조작체계입력재촉(prompt)에로 되돌아 오는 체계를 들수 있다. 사용자는 다른 조작체계호출의 실행을 새치기함으로써 관리자방식으로 들어 올수 있다. 그러므로 프로그람을 실행하기전에 새치기가 적절한 방법으로 처리될수 있도록 새치기표를 설치하는것이 중요하다.

우의 론의로부터 새치기표는 공격자에게 특별히 흥미 있는 점이며 그것을 엄격히 보호하여야 한다는것을 알수 있다. 또한 공격자는 새치기벡토르를 바꾸어 쓰는 방법으로 조작체계의 완정성을 쉽게 약화시킬수 있다. 이러한 문제들은 콤퓨터비루스를 론의할 때 다시 취급될것이다.

4. Motrola 68000에서의 보호

Motorola 68000은 16bit처리기이다. 그것은 16bit의 상태등록기를 가지는데 그의 웃바이트를 체계바이트라고 하며 거기에는 다음과 같은 보안조종관련비트들이 포함된다.

T: 추적비트. 15번비트

S: 관리자비트. 13번비트

$I_2I_1I_0$: 새치기준위번호.8-10번비트

상태등록기의 낮은 바이트는 조건코드등록기(CCR)이다(그림 5-5). 68000은 관리자비트 S를 리용하여 사용자방식과 관리자(체계)방식을 구별한다. 재설정후 68000은 항상 관리자방식에서 시동한다. 관리자방식은 상태등록기의 체계바이트에로의 접근을 허락한다. 일단 S비트가 0으로 설정되여 사용자방식이 능동으로 되면 새치기와 오유례외에 의해서만 S비트를 1로 다시 절환하고 관리자방식에서 동작할수 있다. 조작체계호출은 TRAP #n명령을 통하여 실현되며 관리자방식에서 동작할수 있고 관리자기억기에 대한 접근을 가진다. 연산수# n은 32와 47사이의 례외벡토르를 가리킨다. 여기서 탈퇴할 때 조작체계호출은 RTE명령을 써서 사용자방식에로 되돌아 간다.

68000은 7준위의 새치기우선권을 가진다. 현재새치기의 준위는 상태등록기의 $I_2I_1I_0$ 비트들에 보관된다. 68000이 새치기를 처리하고 있을 때 또 다른 새치기가 보다 높은 우선권을 가지고 발생하면 그때는 새로운 새치기가 우선권을 차지하며 첫번째 새치기는 새치기된다. 낮은 우선권을 가진 새치기는 무시된다. 새치기처리기를 다른 새치기들로부터 막기 위하여서는 상태등록기에서 새치기준위번호를 설정하여 새치기들을 마스크하여야 한다. 그러므로 새치기준위 7은 모든 다른 새치기들을 마스크한다.
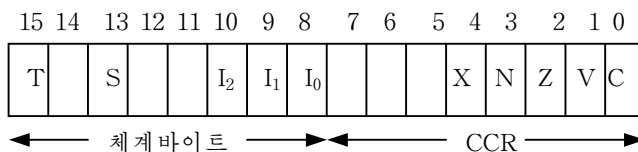
그림 5-5. 68000 상태등록기

처리기는 기억할당식 I/O를 리용한다. 즉 입력/ 출력포구들이 기억주소공간의 부분처럼 취급된다. 이리하여 입력/ 출력조작들과 기억기접근은 단일한 방법으로 취급될수 있다. 처리기는 64개 핀을 가진다. 그중 세개는 기능코드출력, FC_2, FC_1, FC_0으로 설계되였다. 이 기능코드들은 주소해신기에 대한 처리기상태를 나타내는데 이것은 해신기가 사용자기억기와 관리자기억기사이 또는 자료와 프로그람사이를 선택하도록 리용될수 있다(그림 5-6).

자료와 프로그람을 구분하는 능력은 매우 쓸모 있는 보안특성이다. 그것은 프로그람을 수정하는것을 막기 위한 기초이다.

보다 추상적인 관점으로부터 기억기를 여러 구역들로 나눌수 있다. 이때 접근조종은 자료객체나 프로그람이 오는 위치를 참조할수 있다.

FC2	FC1	FC0	
0	0	0	(정의되지 않음,예약)
0	0	1	사용자자료
0	1	0	사용자프로그람
0	1	1	(정의되지 않음, 예약)
1	0	0	(정의되지 않음, 예약)
1	0	1	감시자료
1	1	0	감시기프로그람
1	1	1	새치기응답확인

그림 5-6. MC68000기능코드출력

이제 microcosmos 에서 위치-기호접근조종에 대한 실례를 보자. 분산체계나 콤퓨터망에서 자주 microcosmos 에서의 위치-기호접근조종을 요구한다.

5. Intel 80386/80486에서의 보호

Intel 80386/80486은 32bit 극소형처리기들이다. 80386/80486에서의 보호방식들은 다중과제조작체계들의 완정성과 기밀성요구들을 지원한다. Intel 80386/80486은 상태등록기에 4개의 특권준위들(보호고리들)을 정의하는 2bit 마당을 가진다. 특권준위는 단일명령(PDPF)에 의해서만 변경될수 있는데 이 명령은 준위 0에서 실행되여야 한다. 쏘프트웨어는 이 준위들에 다음과 같이 할당될수 있다.

0: 조작체계핵심부
1: 조작체계의 나머지부분
2: I/O구동기 등
3: 응용쏘프트웨어

모든 조작체계들이 이 4개 준위들을 다 리용하는것은 아니다. 실례로 Unix는 준위 0과 3만을 리용한다. Intel 80386/80486은 다음과 같은 보안방책을 실현한다.

규칙: 수속들은 자기가 속한 고리 또는 바깥고리들에 있는 객체들에만 접근할수 있다. 수속들은 자기가 속한 고리내에서만 부분루틴들을 호출할수 있다.

Intel 80386/80486은 기억기토막들, 접근조종모듈, 서술자들에 있는 문(gate)들과 같은 체계객체들에 대한 정보를 기억한다. 서술자들은 서술자표에 기억되며 선택자들을 통하여 접근된다. 어떤 객체의 특권준위는 그의 서술자의 DPL마당에 기억된다. 선택자는 서술자표에서 객체의 기입항목과 지적하는 색인과 요구특권준위(RPL)마당을 포함하는 16bit마당이다(그림 5-7).
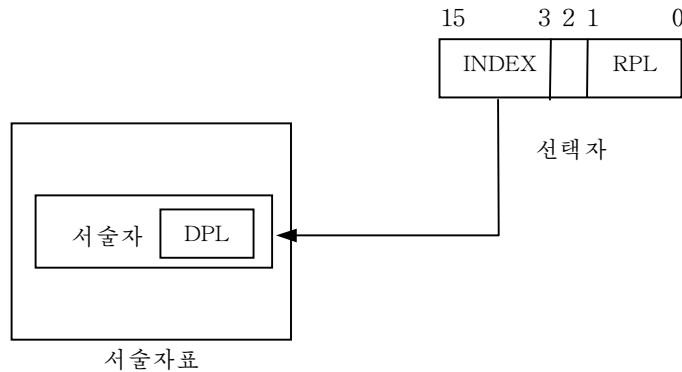
그림 5-7. 선택자의 서술자

RPL마당의 리용은 간단히 설명된다. 조작체계만이 선택자들에 접근할수 있다.

주동체들 례하면 처리들에 대한 정보를 포함하는 체계객체들은 물론 서술자들과 선택자들을 가진다. 주동체가 객체에 대한 접근을 요구할 때 관계되는 선택자들은 해당한 토막등록기들에 넣어 진다. 실례로 현재특권준위(CPL)라고 하는 현재처리의 특권준위는 코드토막(CS)등록기에 기억된 선택자의 특권준위이다.

여기서 또다시 보다 높은 특권을 요구하는 조작들에 대한 접근을 관리해야 하는 문제에 맞다들게 된다. 고리 3에 있는 응용프로그람이 고리 1에 있는 조작체계루틴으로부터 이 봉사를 요구한다고 가정하자. 80386/486에서 이 문제는 문(gate)을 리용하여 해결된다. 문이란 어떤 수속(일부 코드토막에 있는)을 지적하는 체계객체로서 자기가 지적하는 코드토막과 다른 특권준위를 가진다. 문들은 보다 안쪽 고리에 있는 수속에 대한 실행에만 가능한 접근을 허락한다. 바깥쪽으로의 호출들에 대한 제한은 여전히 시행된다.

어떤 수속이 문을 리용하자면 문은 그 수속과 같은 고리안에 있어야 한다. 문을 통하여 부분루틴을 호출할 때 현재특권준위는 문이 지적하는 코드의 준위로 변한다. 부분루틴으로부터 귀환할 때 특권준위는 호출하는 수속의 특권준위로 회복된다. 부분루틴호출 역시 호출하는 수속의 상태와 귀환주소를 가리키는 정보를 탄창에 보관한다. 탄창의 적절한 특권준위를 결정하자면 호출하는 수속은 보다 안쪽 고리에 쓰기를 할수 없다는것을 명심해야 한다. 그러나 바깥쪽 고리에 탄창을 두는것은 귀환주소를 얼마 보호되지 않은것으로 남겨 두기때문에 보안의 견지에서는 오히려 불만족스러운것이다. 그러므로 탄창의 부분(문의 서술자에 아무리 많이 서술된다 해도)은 보다 높은 특권을 가진 탄창토막에 복사한다.

바깥쪽 고리수속이 안쪽 고리수속을 호출하도록 허락함으로써 잠재적인 보안엿보기구멍(loophole)을 만든다. 바깥쪽 고리수속이 안쪽 고리에 존재하는 객체를 바깥쪽 고

리에 복사하기 위해 안쪽 고리수속을 요구할수 있다. 이것은 보안방책을 위반하지 않으므로 어떤 기구로도 방지할수 없다. 이로부터 초기의 보안방책을 현재특권준위뿐아니라 호출하는 수속의 준위도 고려할수 있도록 확장하여야 한다. 80386/486에서는 이와 같은 방책이 선택자의 RPL마당과 요구특권준위조정(ARPL)명령에 의해서 지원될수 있다. ARPL명령은 모든 선택자들의 RPL마당들을 호출하는 수속의 CPL로 고친다. 다음체계는 RPL(선택자에 있는)과 DPL(서술자에 있는)를 비교하여 만일 같지 않으면 요구된 조작을 거부할수 있다(그림 5-8).
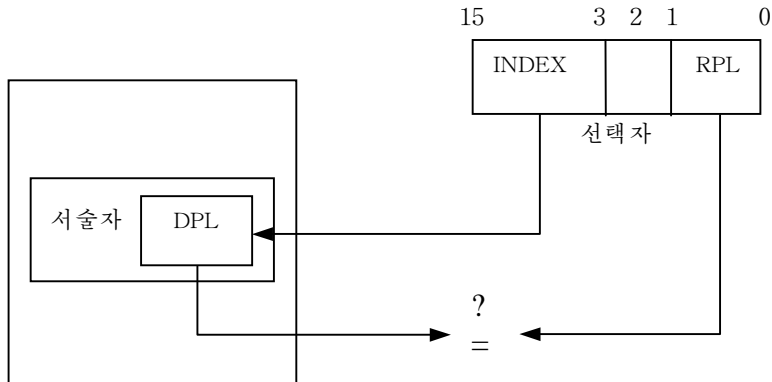
그림 5-8. RPL과 DPL을 비교

제4절. 참조감시기

조작체계들은 자료와 자원들에 대한 접근을 관리한다. 그것들은 보통 사용자자료의 해석에는 관계하지 않는다. 다중과제조작체계들은 서로 다른 사용자들에게 속하는 처리들의 실행을 엇끼워 준다. 이때 조작체계들은 자기의 완정성을 보존해야 할뿐아니라 사용자들이 우연적으로 혹은 고의적으로 다른 사용자들의 자료에 접근하는것을 막아야 한다. 조작체계 그자체의 완정성은 조작체계공간으로부터 사용자공간을 분리하는것에 의해 보호된다. 사용자들의 론리적인 분리는 사용자들사이의 우연한 혹은 고의적인 간섭을 막는다. 분리는 다음의 두가지 준위에서 일어 날수 있다.

- 론리적인 기억기객체들을 처리하는 파일관리
- 물리적인 기억기객체들을 처리하는 기억기관리

보안과 관련되는 한 이 구분은 중요하다. 실례로 기억기를 구성하는 두가지 주요방법으로서 토막화와 페지화를 고찰해 보자. 토막화는 자료를 론리적인 단위들로 나눈다. 매개 토막은 유일한 이름을 가지며 그 토막의 항목들은 토막이름과 그 토막안에서의 해당한 변위를 줌으로써 주소가 지정된다. 기입항목들을 가진다. 조작체계는 기억기에서의 실제주소를 가지는 토막이름표를 보존한다. Multics조작체계는 론리적접근조종을 위해 토막화를 리용한다.

토막화는 론리적단위들로의 분리인데 이것은 보안방책을 실행하기 위한 좋은 기초이다. 한편 토막들은 가변길이를 가지는데 이것은 기억관리를 보다 어렵게 한다.

페지화는 기억기를 같은 크기의 페지들로 나눈다. 주소들은 다시 두개의 부분 즉 페지번호와 그 페지내에서의 변위로 이루어 진다.

페지화는 그것이 효과적인 기억관리를 할수 있게 하므로 널리 리용된다. Multics에서의 토막들은 실제로 페지화된다. 한편 페지화는 페지들이 론리적인 단위가 아니므로 접근조종을 위한 좋은 기초는 아니다. 하나의 페지는 서로 다른 보호를 요구하는 객체들을 포함할수 있다.

나쁜것이지만 유감스럽게도 페지화는 잠복통로를 열수 있다. 론리적객체들은 페지경계를 넘어서 기억될수 있다. 이런 객체가 접근될 때 조작체계는 일정한 단계에서 새로운 페지를 요구할것이며 페지오유가 일어 날것이다. 만일 페지오유가 관찰될수 있으면 대부분 조작체계들에서처럼 사용자는 접근요구에 대한 적절한 결과를 초과하는 정보를 제공받게 된다.

실례로 통과암호기구를 고찰하자. 사용자는 통과암호를 입력하며 이것은 한문자씩 주사되여 기억기에 기억된 참고통과암호와 비교된다. 맞지 않는 개소가 발견되는 순간에 접근은 거부된다. 만일 통과암호가 페지경계를 넘어서 기억되면 이때 공격자는 페지오유를 관찰함으로써 첫번째 페지에 있는 통과암호의 쪼각이 정확히 추측되였다고 추론할수 있다. 만일 공격자가 통과암호가 기억된 페지를 조종할수 있다면 통과암호의 추측은 오히려 쉽게 된다.

1. 기억기보호

조작체계가 자기의 완정성을 보호하고 매개 처리가 분리된 주소공간에만 한정되기를 원한다면 우리의 과제들중의 하나는 기억기에 있는 자료객체들에 대한 접근을 조종하는 것이다. 이러한 자료객체는 물리적으로 정해 진 기억위치들에 기억된 비트들의 집합으로 표현된다. 론리적객체에 대한 접근은 최종적으로 기계언어준위의 접근조작들로 변환된다. 이 준위에서 기억위치들에 대한 접근을 조종하기 위하여 다음과 같은 세가지 선택방안이 주어 진다.

- 조작체계가 사용자처리들로부터 받은 주소를 수정한다.
- 조작체계가 사용자처리들로부터 받은 상대주소를 유효주소로 바꾼다.
- 조작체계가 사용자처리들로부터 받은 주소가 주어 진 한계내에 있는가를 검사한다.

주소모래통처리(sandboxing)는 첫번째 방법에 대한 실례이다. 주소는 토막식별자와 변위부로 구성된다. 조작체계가 주소를 받으면 정확한 토막식별자를 설정한다. 그림 5-9는 이것이 두개의 등록기조작으로 어떻게 수행되는가를 보여 준다. 우선 주소와 마스크 _1과의 비트별 론리곱하기(AND)는 토막식별자를 지운다. 다음 마스크 _2와의 비트별 OR는 토막식별자를 예견된 값 SEG_ID로 설정한다.

두번째 방법에서는 주소화방식을 리용하여 금지된 기억구역밖의 처리들을 보존한다. 만일 주소화방식에 대한 더 많은 지식이 요구되면 조작체계나 콤퓨터구성방식에 대한 책들[38, 66, 100]을 참고하시오. 여러가지 주소화방식들중에서 상대주소화는 특별한 흥미를 끄는데 여기서 주소는 주어 진 기준주소에 대한 상대적인 변위로 표시된다. 실례로 상대주소화는 Motorola 68000에서 변위부를 가진 등록기간접주소화방식을 통하여 지원된다. 아쎔불리언어명령

MOVE .L8(A1), D3

은 기억기주소를 얻기 위하여 등록기 A1의 내용에 상수 8을 더한다. 이 기억기주소에 기억된 긴 단어(16bit)는 등록기 D3에로 옮겨 진다.
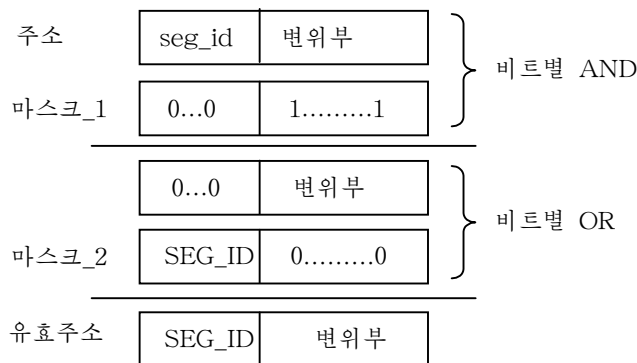
그림 5-9. 주소계산

상대주소화는 위치독립형코드화를 허용한다. 그래서 프로그람은 기억기 어느 곳에든지 기억될수 있으며 기억관리프로그람에 보다 큰 유연성을 준다. 그것은 또한 경계등록기들의 리용을 편리하게 한다. 경계등록기는 조작체계에 할당된 기억구역의 끝의 주소를 포함한다. 사용자프로그람에서의 주소는 상대주소로 해석된다. 조작체계는 다음 유효주소를 얻기 위하여 경계등록기에 관한 상대주소화를 리용한다(그림 5-10). 이런 방법으로 사용자프로그람은 조작체계공간의 밖에 위치하고 있는것들에만 접근할수 있다. 서로 다른 사용자들에게 할당된 기억구역들을 분리하기 위하여 류사한 방법들이 조작체계에서 사용될수 있다.

이 방법은 기준등록기들과 한계등록기들을 통하여 처리에 할당된 기억공간을 확정함으로써 보다 세련될수 있다. 한걸음 더 나가서 사용자프로그람공간과 자료공간을 위한 기초 및 한계등록기들을 각각 도입한다. 이런 기구의 적절한 리용을 위해 처리기는 주어진 기억위치가 자료나 프로그람코드를 포함하는가를 검출할수 있어야 한다. 그러나 대부분 명령모임들에는 자기들의 연산수들의 형을 검사하기 위한 수단들이 없다. 그것들이 없을 때 형정보는 서로 다른 기억접근조작들에서 리용되여야 하는 주소등록기를 지정함으로써 프로그람에서 제공될수 있다. 이것은 적절한 프로그람작성숙련을 요구한다.

한편 표쪽 붙은 구성방식(tagged architecture) 에서는 매개 자료항목이 그의 형을 나타내는 표쪽을 가진다. 조작전에 CPU는 기억기에 기억된 값으로부터 어떤 형위반도 직접 검출할수 있다.이 표쪽들은 보안방책을 시행하는데도 리용할수 있다. 력사적으로 표쪽 붙은 구성방식들은 실제적실현에서보다 리론적고찰에 많이 쓰였다. 그 몇가지 실례로 burroughs B6500-7500체계와 IBM system/38[17]을 들수 있다(콤퓨터의 력사에 흥미를 가지는 독자를 위해서 폰 노이만(Von Neumann)은 자기가 1945년에 쓴 First Draft of a Report on the EDVAC에서 표쪽 붙은 구성방식을 언급하였다[154]).
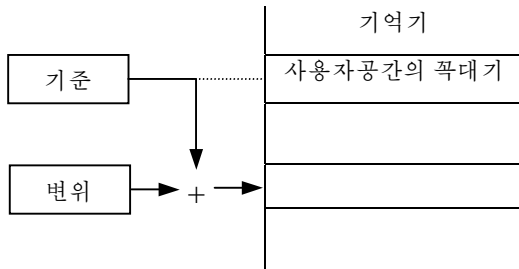
그림 5-10. 기준등록기의 주소화

그림 5-11. 표쪽 붙은 구성방식

그림 5-11은 기억기객체들 례하면 옹근수(int), 문자렬(STR), 조작수(OP)의 형을 나타내는 표쪽 붙은 구성방식(tagged architecture)을 보여 준다. 표쪽(tag)들은 어떤 접근조작이 기억기위치에서 수행될수 있는가 례하면 읽기, 쓰기, 실행 등을 나타내기 위해서도 리용될수 있다.

2. Multics

Multics조작체계의 력사적실례는 보안핵심이 어떻게 접근조종을 시행할수 있는가를 보여 주고 있다. Multics는 안전하고 믿음직한 다중사용자조작체계로 개발되였다[13, 119]. Multics 에 의하여 BPL과 같은 보안에 관한 많은 연구들이 진행되였다. Multics에서의 보호기구들에 대한 개괄을 [119]와 4장에서 주었다. Multics는 방대한 목표들과 보안요구들을 제기하였으므로 그것을 실현하는것은 너무 힘이 부치였다. 이로부터 훨씬 더 간단한것 즉 Unix가 창안되였다.

3. BLP의 Multics해석

Multics에 대한 연구는 보안모형, 이 경우에 벨-라파둘라모형이 안전한 조작체계의 설계에 어떻게 리용되는가를 볼수 있게 한다. 접근조종의 공식적모형으로서 BLP는 조작체계들의 보안요구들을 취급하는데 매우 편리하다. 사실상 그것은 바로 그런 목적을 위해서 개발되였다. BLP에서 유도된 보안정의는 상대적으로 안전보안체계를 구축하기 쉽

게 한다. 다만 보안성을 담보하기 위하여 상태이행들을 적절하게 정의해야 한다. Multics가 안전하다는것을 보증하기 위해서는 BLP로 구성한 Multics의 묘사를 찾아야 한다. BLP의 개념들이 어떻게 Multics에서 사용되는가를 보여 주기 위해 대체로 [13]에 주어 진 표현을 따른다.

Multics에서 주동체들은 처리들이다.매개 주동체들은 처리에 대한 정보를 포함하는 하나의 서술자토막을 가진다.여기에는 그 처리가 현재 접근하고 있는 객체에 대한 정보가 포함된다.이 매개 객체들에 대하여 주동체의 서술자토막에는 하나의 토막서술자단어(SDW)가 대응된다. SDW의 쓰기법은 그림 5-12에 주어 진다. SDW는 객체의 이름, 객체에 대한 지시자 그리고 읽기, 실행, 쓰기접근을 위한 지시기기발들을 포함한다. 이 지시기들은 제3장 3절 2에서 서술한 접근속성들을 따른다. 주동체들의 보안준위들은 처리준위표와 현재준위표에 보존된다. 능동토막표는 모든 능동처리들의 경로를 보존한다. 능동인 처리들만이 객체에 대한 접근을 가진다.

<table>
<tr><td colspan="4">토막-id</td><td>지시자</td><td></td></tr>
<tr><td></td><td>r:on</td><td>e:off</td><td>w:on</td><td colspan="2"></td></tr>
</table>

그림 5-12. Multics 토막서술자단어

Multics에서 객체들은 기억기토막들, I/O 장치들 등이다. 객체들은 등록부나무에서 계층적으로 조직된다. 등록부들 역시 토막이다. 어떤 객체의 보안준위나 그의 접근조종표(ACL)과 같은 객체에 대한 정보는 그 객체의 어미등록부에 보존된다. 객체의 접근조종파라메터들을 변경시키거나 객체를 창조하고 삭제하는것은 그의 어미등록부에 대한 쓰기 혹은 추가접근권한을 요구한다.

객체에 접근하기 위해서 처리는 목적객체에 대한 뿌리등록부로부터 등록부나무를 횡단하여야 한다. 만일 이 경로에 그 처리가 접근할수 없는 등록부가 있다면 목적객체에 접근할수 없다. 다른 말로 비밀등록부안에 있는 비밀이 아닌 객체를 비밀이 아닌 사용자가 읽어 낼수 없다. 따라서 객체들을 보다 높은 보안준위를 가지는 등록부에 배치하는것은 좋지 않으며 언제나 객체의 보안준위가 그의 어미등록부의 보안준위우에 놓일것이 요구된다. 이 속성을 호환성이라고 한다. Unix와 같은 형태의 조작체계들에서도 같은 문제를 취급한다. 만일 다른 사용자들이 리용가능한 파일을 만들려 한다면 등록부경로에 접근조종권한을 설정하여야 한다.

Multics체계표들에 있는 자료와 서술자토막을 가지는 BLP상태모임의 요소들을 확인하는데 필요한 모든 정보를 보기로 하자.

- **현재 접근 b**: 능동처리의 서술자토막에 있는 SDW들에 기억된다. 능동처리들은 능동토막표에서 찾는다.

- **접근조종행렬 M**: ACL들에 의해서 표현된다. 매개 객체에 대하여 ACL은 그의 어미등록부에 기억된다. 매개 ACL항목에는 처리와 그 처리가 객체에 대하여 가지는 접근권한들이 지정된다.
- **준위함수 F**: 주동체들의 보안준위들은 특수한 보안준위표(처리준위표와 현재준위표)에 기억된다. 객체의 보안준위는 그의 어미등록부에 기억된다.

제3장 3절 2에서 이미 자료토막과 등록부토막을 위한 Multics접근속성들이 소개되였다. 거기서 이 접근속성들이 벨-라파둘라모형의 접근권한들과 어떻게 대응하는가를 설명하였다. 그림 5-13에 있는 자료토막들을 위한 접근속성들을 다시 서술한다.

접근속성	접근권한
read	r
execute	e, r
read. write	w
write	a

그림 5-13. 자료토막을 위한 접근권한

BLP보안속성들은 처리들과 자료토막들 그리고 SDW들에 기억된 지시기들의 보안준위의 용어들로 바꾸어 표현된다. 실례로 *-속성은 다음과 같이 씌여 진다.어떤 능동처리의 서술자토막에 있는 임의의 SDW에 대해서 처리의 현재준위는

- 만일 읽기나 실행지시기가 ON이고 쓰기지시기가 OFF라면 토막의 준위를 지배한다.
- 만일 읽기지시기가 OFF이고 쓰기지시기가 ON이면 토막의 준위에 의해서 지배된다.
- 만일 읽기지시기가 ON이고 쓰기지시기가 ON이면 토막의 특권준위와 같다.
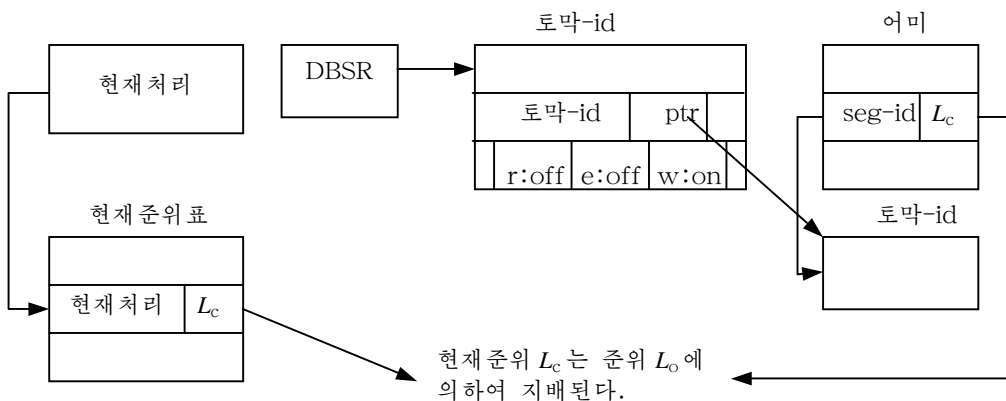
그림 5-14. 접근속성쓰기를 위한 *-속성

그림 5-14는 *-속성을 가지는 허락이 어떻게 증명될수 있는가를 보여 준다. 현재처리의 보안준위 LC는 현재준위표에 기록된다. 서술자토막기초등록기(DSBR)의 내용은 현재처리의 서술자토막의 머리부를 가리킨다. 이 서술자토막은 접근속성이 Write Only인 객체를 위한 SDW를 포함하도록 일어 난다. 따라서 쓰기지시기는 ON이고 읽기지시기는 OFF이다. 객체의 보안준위 L_0은 그의 어미등록부로부터 취해 지며 $L_S \geq L_0$임을 검사하기 위해 L_S와 비교된다.

4. 핵심부의 원시조작들

끝으로 핵심부원시조작들의 모임이 서술되여야 한다. 이 핵심부원시조작들은 Multics핵의 추상적모형에서의 상태이행들이며 우리는 그것들의 BLP보안방책들을 준수한다는것을 보여 주어야 한다. 다음에 기본보안리론의 전제조건들을 세우고 Multics의 《보안》을 증명한다. 또한 핵심부원시조작들의 실현이 그리고 끝으로 주어 진 하드웨어 가동환경에서 그것들의 실행이 그자체의 서술에 따른다는것을 보여 주어야 한다.

핵심부원시조작을 자세히 보기 위해 Get-read 를 선택한다. Get-read원시조작은 그의 파라메터로 Process-ID와 Segment-ID를 취한다. 조작체계는 다음의 사실들을 확인하여야 한다.

- 토막의 어미등록부에 기억된 Segment-ID의 ACL은 읽기허가를 가지는 Process-ID의 목록을 받는다.
- Process-ID의 보안준위는 Segment-ID의 보안준위를 지배한다.
- Process-ID는 신용받는 주동체이거나 또는Process-ID의 현재보안준위가 Segment-ID의 보안준위를 지배한다.

만일 이 세가지 조건들이 다 맞는다면 접근이 허락된다. 만일 Segmetn-ID를 위한 SDW가 존재하지 않으면 대응하는 SDW가 활성화된 읽기지시자와 함께 Process-ID의 서술자토막에 추가된다. 만일 Segment-ID를 위한 SDW가 이미 Process-ID의 서술자토막에 존재한다면 이 SDW에 있는 읽기지시기가 On으로 된다. 만일 세가지 조건들중 어느 하나라도 맞지 않으면 접근은 거부된다.

다음의것들은 Multics핵심부에서 실현하기 위해 제안되였던 일련의 기타 원시조작들이다.

release-read　처리가 객체를 해방한다. 대응하는 SDW 에 있는 읽기기발이 내려 진다. 만일 그후에 설정된 지시기가 없으면 그 SDW는 서술자토막에서 제거된다.

give-read　처리가 다른 처리에게 읽기접근을 넘겨 준다(자유접근조종).

rescind-read　처리가 다른 처리에 주었던 읽기허가를 회수한다.

create-object 처리가 객체를 창조한다. 조작체계는 그 객체의 등록부토막에 대한 쓰기접근이 허락되는가와 그 토막의 보안준위가 처리의 특권준위를 지배하는가를 확인해야 한다.

delete-object 객체를 삭제할 때 create-object에서와 같이 확인을 한다.

change-subject-current-security-level 조작체계는 변경에 의해 보안위반이 발생하지 않는가를 확인해야 한다. 이 핵심부원시조작뿐아니라 원시조작 change-object-current-security-level 도 실현에 대해서는 예견하지 않았다 (안정성).

리상적으로 처리기들은 자기들의 명령모임들이 조작체계의 핵심부원시조작들과 밀접히 련관되도록 개발된다. 반대로 핵심부원시조작들은 현존하는 처리기들이 제공하는 지원을 받을수 있게 설계할수 있다.

이 장의 문헌안내

기초적인 접근조종실례들은 문헌 [82]에 주었다. 첫 다중사용자조작체계에서 목적한 콤퓨터보안결과는 [39]에서 포괄적으로 보여 주었다. 기타 보안기술들에 대해서는 [84]에서 개괄하였다. 안전한 다중사용자조작체계의 설계에 리용된 기술들에 대한 책은 [56]이다. 이 책은 이 분야에 대한 많은 쓸모 있는 참고문헌들을 지적하고 있다. Multics보안과 특히는 보안관리의 복잡성과 설계의 정확성을 평가하는 복잡성해석은 [131] 에서 찾아 볼수 있다.

Motorola 68000 극소형처리기는 [33]에 있는 실례에서 상세히 설명하였다. Intel 80386/486 에서의 보호기구들에 대한 상세한 서술은 [111]에 주었다. Intel 80x86처리기들을 가지고 보안핵심을 구축하는것은 [141]에서 론의되였다.

주소모래통처리과 그와 관련된 기술들은 [155]에서 론의된다. 기억관리에 대한 실천적자료들을 다음의 Web페지를 참고하시오.

http://www.com/literature/ntosysarch/ntosysarch.html

Multics에 대한 정보는

http://www.lilli.com/multics.html

에서 찾아 볼수 있다. 안전한 콤퓨터개발의 초기력사는 ［93］에 묶어 져 있다. 최근에 TCB들(보안핵심)이 아직도 안전체계를 구성하는데서 적절한 모형으로 되는가 하는 의문이 제기되였다. 이 론의에서 대치되는 점들이 ［21］(반대)과［11］(찬성)에 주어졌다.

련습문제

1. Motorola68000에서 상태등록기에 있는 추적비트가 설정되면 매개 명령이 실행된후에 례외(새치기)가 발생하며 프로그람작성자로 하여금 례외조종루틴(새치기처리기)을 실행할수 있게 한다(례로 오유수정프로그람). 추적비트를 설정하는것은 보안과 어떤 관련이 있는가 ? 례외를 처리할 때 왜 추적비트를 지워야 하는가?

2. 기생적인 비루스는 실행가능한 프로그람들을 감염시킨다(제8장 8절을 보시오). 프로그람과 자료를 구별하는 능력이 이러한 비루스들에 대한 방어를 구축하는데 어떤 도움을 줄수 있는가?

3. 지능카드들에서 극소형처리기들은 자기의 전체 조작체계를 ROM에 가지고 있다. 현재조작체계의 일부를 EP-ROM으로 내리적재할수 있는 극소형처리기방향으로 향하고 있다. 조작체계를 ROM에 보관하는것이 어떤 우점과 결함이 있는가? 조작체계의 일부를 EP-ROM에 옮기는것은 보안과 어떤 관련이 있는가?

4. 제5장 4절 3에서 *-속성의 재표현(rephrasing)을 정당화하시오. 왜 세가지 경우를 고려해야 하는가?

5. 여러준위안전조작체계의 뿌리등록부에는 어떤 표식이 할당되여야 하는가?

6. 여러준위안전조작체계에서의 등록부를 고찰하자. 여기에는 파일의 안전준위를 기억하는 다음과 같은 세가지 선택방안이 있다.
   1) 등록부에 파일이름과 함께 파일의 보안준위를 기억한다.
   2) 등록부는 파일의 이름만 포함하고 파일의 보안준위는 파일 그자체와 함께 기억된다.
   3) 등록부를 매개 보안준위용구획(partition)들로 나눈다. 파일이름들은 그의 보안준위에 대응하는 구획에 넣어 진다.
   이상의 3가지 선택방안들 각각의 보안 및 리용가능성련관을 해석하시오.

7. 보안핵심이 없이 보안을 가질수 있는가? TCB와 같이 보안핵심을 가지는것의 우점과 결함을 말하시오.

8. 임무들의 구분, 추상자료형, 원시조작과 같은 세가지 원리들이 안전체계를 구축하는데 어떻게 적용되는가를 보여 주는 실례들을 들어 보시오(원시조작은 보안을 유지하기 위해 단번에 수행되여야 한다. 만일 그것이 새치기되면 체계는 불안정한 상태로 떨어 질수 있다).

9. 어떤 체계에 가입할 때 사용자는 자기의 신원(Identity)을 가지고 실행되는 처리를 시작한다. 사용자가 체계에서 탈퇴(logged off)한후에도 리용자의 신원을 가진 처리들이 실행을 계속할수 있는가? 자신이 리용하는 조작체계에서 이 문제를 조사하시오.

제2편. 실 천

제6장. Unix 보안

지금까지는 개별적인 보안기구들을 고립적으로 보았다. 그러나 실천에서는 그것들이 서로 의존한다. 실례로 접근조종과 인증은 서로 협동해야 하며 서로 다른것이 없이는 효력을 내지 못할것이다. 여기서는 조작체계가 제공하는 보안기구에 주의를 돌리기로 한다.

여기서는 Unix를 실례로 들어 보안기구에 대하여 구체적으로 조사한다.

목적

- 전형적인 조작체계들이 제공하는 보안특징들을 파악한다.
- Unix 보안의 기초를 소개한다.
- 일반보안원리들이 실제의 체계에서 어떻게 실현되는가를 본다.
- 부단히 변하는 환경에서의 보안관리과제를 정확히 인식한다.

제1절. 소개

조작체계들은 정연한 보안조종모임을 제공하기 위해 식별과 인증, 접근조종과 검열(auditing)과 같은 요소블로크(building block)들을 결합한다. 유연하고 《특징이 풍부한》 보안방책을 제공하려고 한다면 보안기구들은 점점 더 복잡해 진다. 이런 환경에서 TCB는 너무 커서 《핵심부(kernel)》에 적합치 않다. 여기서는 앞에서 본 계층모형의 조작체계준위에서 제공하는 보안조종들을 본다(그림 6-1).

조작체계의 보안을 평가할 때 다음과 같은 문제들이 제기된다.

- 어떤 보안특징들이 실현되여 있는가?
- 이 보안특징들을 어떻게 관리할수 있는가?
- 보안특징이 유효하다는 담보가 있는가?

널리 보급된 조작체계들에는 보안조종을 조직하는 일반적인 형식이 있다. 사용자(주동체)들에 대한 정보는 사용자등록자리에 기억된다.

사용자에게 부여된 임의의 특권들은 이 등록자리에 기억되게 된다. 식별과 인증은 사용자의 신원을 확인하는것인데 이것은 체계가 사용자의 특권과 사용자에 의해 시작되는 어떤 처리를 련결시키도록 한다. 자원(객체)에 대한 허락은 체계관리자 또는 그 자원의 소유자에 의해 설정된다. 조작체계는 어떤 사용자의 접근요구를 허가할것인가 아니면 거절할것인가를 결정할 때 사용자의 신원, 사용자의 특권 그리고 그 객체에 대한 허가들을 참고할수 있다.
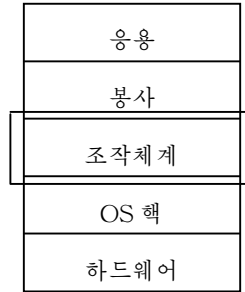
그림 6-1. 조작체계에서 보안

보안은 권한이 없는 작용들을 막을뿐아니라 이러한 작용들을 검출도 한다. 우리는 공격자들이 보안기구주위에서 맴돌수 있다는 사실에 주의해야 한다. 보안위반을 조사하든가 공격시도를 추적할수 있도록 사용자들이 진행한 행동리력을 보관하는 대책을 취해야 한다. 때문에 조작체계는 보안관련사건들의 검열기록(검열궤적)들을 보관하고 보호하여야 한다.

끝으로 조작체계에 최상의 보안특징들이 있다 해도 그것을 적절하게 리용하지 않는다면 그 특징들은 가치를 잃어 버리게 된다는것을 강조한다. 체계는 안전한 상태에서 기동하여야 하며 따라서 조작체계의 설치와 구성은 매우 중요한 문제로 된다. 적당치 못한 기정설정은 보안을 약화시키는 주요한 원인으로 될수 있다. 조작체계는 매우 복잡하고 부단히 갱신되는 쏘프트웨어체계이며 따라서 새로운 갱신판마다 새로운 약점이나 결함이 발견되군 하는것이 보통의 일이다. 때문에 각성 있는 체계관리자들은 CERT(콤퓨터비상사태대응팀-computer emergency response teams)의 보안에 관한 권고를 참작하여 현재의 수준에서 크게 비약하지 말아야 한다.

조작체계보안의 골격에 대하여 다음과 같은 순서로 보기로 한다.

- 가입과 사용자등록자리
- 접근조종
- 검열(audit)
- 구성 및 관리

이 장에서는 Unix조작체계의 보안특징들을 조사한다.

Unix는 그의 설계력사로 하여 믿음성이나 보안에서 좋은 평가를 받지 못하였다[102]. 그러나 적절히 리용한다면 아주 효과적인 보안특징모임을 제공한다. Unix에는 일부 전문사항들과 보안조종을 실행하는 수법들에서 차이나는 여러가지 판본들이 있다. 따라서 Unix보호를 규격화하려는 시도들이 제기되였다. Unix체계와의 공통적인 대면을 정의하는 규격인 POSIX 1003계렬에서 POSIX 1003.6이 보안공정들을 취급한다. 이 장은 Unix보안에 대한 완벽한 소개나 Unix체계보안설치에 대한 지도서가 아니다. 그래서 Unix보안의 기초와 일반적인 관심사로 되는 점들을 례증하는 보안특징의 일부만을 강조하는것으로 그치기로 하였다. 이 책에서 취급하는 Unix보안의 분야는 아직 사람-기계의 척도에서 놓고 볼 때 기계쪽으로 더 치우쳐 있다(그림 6-2).
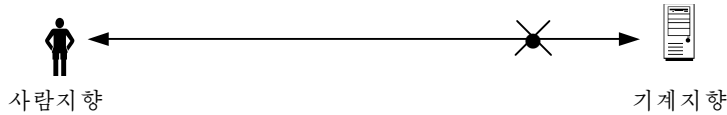
그림 6-2. 사람 - 기계척도에서 Unix 보안

제2절. Unix 보안구성방식

대부분의 안전한 조작체계들은 보안이 어떻게 실시되며 보안관련자료들이 어디에 보관되는가를 설명하는 보안구성방식(architecture)을 가지고 있지만 Unix는 판본들을 가지치기하거나 집중시켜 온 력사를 가지고 있다. 이것은 원래의 설계목적이라기보다 요구가 제기될 때마다 필요한 보안특징들을 Unix에 추가하군 하였다는것을 보여 주고 있다. Unix 가 발전함에 따라 새로운 보안조종이 체계에 추가되였으며 이미 있던 조종기능들이 강화되였다. 설계자들은 새로운 특징을 추가할 때 현존하는 Unix구조에 될수록 적게 간섭하도록 하였다.

제3절. 가입과 사용자등록자리

Unix에서 사용자들은 사용자이름에 의하여 식별되며 통과암호에 의하여 인증된다. 많은 Unix체계들에서 통과암호는 8문자까지로 제한된다. 통과암호들은 모두가 령인 블로크를 시작값으로 하고 통과암호를 열쇠로 하여 약간 수정된 DES알고리듬을 25번 반복하는 crypt(3)알고리듬으로 암호화한다. 암호화된 통과암호들은 /etc/passwd 파일안에 보관된다. 이 파일의 기입항목은 다음과 같다.

사용자이름: 암호화된 통과암호 : 사용자ID: 그룹ID: ID문자렬: 홈등록부: 가입쉘

ID문자렬마당은 사용자의 완전이름(full name)을 포함한다. 사용자 ID와 그룹 ID는 이 장의 뒤부분에서 설명한다. 마지막 두개의 마당들은 사용자의 홈등록부와 가입에 성공한 사용자가 리용할수 있는 Unix 쉘들을 명기한다. 체계가 취하는 행동들은 /etc/profile 파일안에 기록된다. 사용자고유의 설정들은 사용자홈등록부안의 .profile안에 정의된다. 사용자가 제일 마지막으로 가입한 시간은 /usr/adm/lastlog파일에 기록되며 finger지령으로 꺼내볼수 있다. cat/etc/passwd나 less/etc/passwd로 통과암호파일을 현시하면 다음과 같은 항목들을 볼수 있다.

dieter:RT.QsZEEsxT92:10026:53:Dieter Gollmann:/home/staff/dieter:
/usr/local/bin/bash

사용자의 통과암호마당이 비여 있으면 사용자는 가입할 때 통과암호를 주지 말아야 한다. 통과암호마당이 《*》기호로 시작되면 사용자는 가입할수 없다. 왜냐하면 평문통과암호에 한방향함수를 적용한 결과는 《*》로 시작될수 없기때문이다. 이것은 사용자의 등록자리를 금지시키는 가장 일반적인 방법이다.

통과암호는 passwd(1)지령으로 변경시킬수 있다. 먼저 낡은 통과암호를 입력할것을 요구한다. 이것은 사용자이외의 다른 사람이 통과암호를 변경시키지 못하게 한다. 통과암호를 입력할 때 문자들이 화면에 나타나지 않으므로 확인하기 위해 새 통과암호를 두번 입력할것을 요구한다. 이때 두번의 입력이 일치해야 한다. 통과암호를 변경시킨후 다시 가입하거나 또는 su(1)(사용자설정)지령으로 변경의 효과를 확인할수 있다.

보안을 의식한 Unix의 판본들은 보다 개선된 통과암호보안대책들도 제공한다. 통과암호는 절임되거나 그림자통과암호파일 /.secure/etc/passwd안에 보관된다. 통과암호선택실습은 약한 통과암호를 사용하지 않도록 도움을 준다. 낡은 통과암호의 재리용을 통제하며 통과암호에 유효기간을 설정할수 있다. 또한 뿌리가입은 /etc/ttys안에서 지정한 말단으로 제한할수 있다. 제3부류의 Unix 보안제품들도 류사한 보안봉사를 제공한다.

1. 사용자와 특권사용자

Unix는 8문자까지의 사용자이름으로 사용자를 나타내며 내적으로는 16bit수의 사용자 ID(UID)로 표현한다. UID들은 /etc/passwd 안에 있는 사용자이름과 련결된다. Unix는 같은 UID를 가지는 사용자들을 구별하지 못한다. 특수한 의미를 담고 있는 일부 UID들을 그림 6-3에 보여 준다.

-2	nobody
0	root
1	daemon
2	uucp
3	bin
4	games
9	audit

그림 6-3. 특수한 사용자ID

매 Unix체계에는 특별한 권한을 가진 하나의 사용자-특권사용자가 있다. 이 특권사용자는 UID 0을 가지며 보통 뿌리(root)라는 사용자이름을 가진다. 뿌리등록자리는 검열기록을 만드는 가입이나 I/O장치호출과 같은 주요한 본질적인 과제들을 위하여 조작체계가 리용한다. 특권사용자에 대해서는 거의 모든 보안검열들이 무시된다. 체계관리파

제를 수행하기 위해서는 뿌리등록자리가 요구된다. 체계관리자는 뿌리를 자기의 개인등록자리로 리용하지 말아야 한다. 뿌리에로의 변경은 사용자이름을 명기하지 않고 필요할 때 /bin/su를 입력하여 요구할수 있다.

특권사용자는 거의 모든것을 할수 있다. 실례로 특권사용자는 임의의 다른 사용자로 될수 있다. 특권사용자는 체계박자도 변경시킬수 있다. 특권사용자는 자기에게 적용된 일부 제한들을 회피할수 있다. 실례로 특권사용자는 읽기만으로 설치된 파일체계에 쓰기할수는 없지만 그 파일체계를 해체하고 쓰기용으로 다시 설치할수 있다. 그러나 특권사용자도 crypt가 한방향함수이므로 통과암호는 해득할수 없다.

특권사용자가 이처럼 많은 권한을 가지는것은 Unix의 주요약점으로도 된다. 특권사용자상태를 획득한 공격자는 전반체계를 쉽게 장악할수 있다. 따라서 특권사용자상태에 대한 접근을 통제하는데 주의를 돌려야 한다. /etc/passwd와 /etc/group 파일들은 쓰기보호되여야 한다. /etc/passwd 를 편집할수 있는 공격자는 자기의 UID를 0으로 변경시킴으로써 특권사용자로 될수 있다. 특권사용자로 되기 위하여 /bin/su를 리용하는것은 조작체계가 어떤 다른 등록부에 삽입된 su의 판본을 참조하는것을 막는다. 검열기록에는 su지령을 제출하는 사용자와 모든 su시도들을 기록하여야 한다. 망을 취급하는 uucp나 daemon과 같은 특수한 사용자들에게 체계관리자의 임무를 분담시킴으로써 그중 어느 한 사용자가 손상되였다고 해도 체계전반이 손상되지 않게 한다.

2. 그룹

사용자들은 한개 또는 그이상의 그룹들에 속한다. 사용자들을 그룹으로 묶는것은 접근조종을 편리하게 한다. 실례로 전자우편에 접근할수 있도록 허락된 모든 사용자들을 mail이라는 그룹로 묶을수 있고 또 operator 라는 그룹안에 모든 조작자들을 넣을수 있다. 매 사용자들은 1차그룹(primary group)에 속한다. 1차그룹의 그룹ID(GID)는 /etc/passwd 안에 보관된다. /etc/group 파일은 모든 그룹들의 목록을 포함한다. 이 파일의 기입항목들은 다음의 형식을 가진다.

그룹이름: 그룹통과암호:GID: 사용자목록

실례로 기입항목

infosecwww:*:209:chez,af

는 infosecwww그룹이 금지된 통과암호를 가지며 GID는 209이고 chez 와 af라는 2명의 성원을 가진다는것을 보여 준다. 그림 6-4는 특수한 의미를 가지는 그룹ID들을 보여준다.

System V Unix 에서 어느 한 시점에서 사용자는 하나의 그룹에만 속할수 있다. 현재그룹은 newgrp지령으로 변경시킨다. 사용자들이 그들이 이미 성원으로 있는 그룹에로 변경하는것은 자유이다. 그러나 사용자들이 자기가 성원으로 속해 있지 않은 그룹에로 변경하려고 시도한다면 newgrp는 통과암호를 요구하고 입력한 그룹통과암호가 정확한 경우에만 림시적인 성원자격을 준다. Berkeley Uinx 에서는 한 사용자가 한개이상의 그룹들에 속할수 있기때문에 newgrp지령이 필요 없다.

0	system/wheel
1	daemon
2	uucp
3	mem
4	bin
7	terminal

그림 6-4. 특수한 그룹ID들

3. 사용자 ID설정과 그룹 ID설정

통제된 호출에 대한 문제를 보기로 하자. Unix에서 어떤 조작체계기능을 실행하려면 특권사용자의 특권을 가져야 하지만 사용자들에게는 특권사용자의 상태를 주지 말아야 한다. 량쪽의 요구를 다같이 맞추는 방법을 찾아 내야 한다. Unix 에서는 이를 위해 사용자 ID설정프로그람(SUID)과 그룹 ID설정프로그람(SGID)을 사용한다. 이 프로그람들은 그 소유자 또는 그룹의 유효한 사용자 ID나 또는 그룹 ID를 가지고 실행시키는데 이때 다른 사용자들이 정상적으로는 호출할수 없는 파일들을 림시적으로 또는 제한적으로 호출할수 있게 해준다. SUID프로그람의 소유자는 뿌리이며 이 프로그람을 실행하는 사용자는 실행기간에 특권사용자의 상태를 얻게 된다. 중요한 SUID프로그람들은 다음과 같다.

/bin/passwd　통과암호변경
/bin/login　　프로그람가입
/bin/at　　　묶음일감제출
/bin/su　　　UID프로그람의 변경

여기에 주의해야 할 문제가 있다. 사용자가 SUID프로그람을 실행하는 동안 프로그람소유자의 특권을 가지기때문에 이 프로그람은 오직 소유자가 의도한것만을 해야 한다. 이것은 뿌리에 소속된 SUID프로그람에 대해서는 특별히 중요하다. SUID프로그람의 실행을 중단시키고 그의 기능을 변경시킬수 있는 공격자는 특권사용자의 자격을 요구하는 행동에 착수하며 공격기간에뿐아니라 다른 기회에도 특권사용자의 자격을 얻을수 있도록 체계를 변경시킬수 있다. 이로부터 위험은 사용자호상작용을 하는 SUID프로그람에서 발생된다고 볼수 있다. 특별한 함정은 특권사용자로서 실행하고 있는 기간에 사용자에게 쉘지령을 호출할수 있게 해주는 쉘확장(shell escapes)이다. 프로그람들은 오직 실제로 필요한 때에만 SUID상태를 가져야 한다. 체계관리자들은 SUID프로그람의 완정성을 특별히 주시해야 한다. SUID프로그람들이 공격에 성공적으로 리용된 두가지 실례연구가 [55]에 있다.

제4절. 접근조종

접근조종은 사용자들의 특성과 파일, I/O 장치, 기억기 등과 같은 자원들의 속성에 기초한다. 표준Unix체계는 소유자, 그룹, 전체(world)의 개념을 도입하여 자유접근조종을 편리하게 한다. 특권사용자들은 이러한 류의 접근조종에 구애되지 않는다. Unix는 파일과 장치들을 구별하지 않으며 모든 자원들을 유일한 방법으로 취급한다.

1. Unix파일구조

Unix는 파일들과 등록부들이 포함되여 있는 나무구조의 파일체계안에 파일들을 배치한다. 등록부안에서 매 파일기입항목들은 이노드(inode)라고 부르는 자료구조에로의 지시자이다. 그림 6-5 는 이노드안에서 접근조종과 관계되는 마당들을 보여 준다. 매 등록부는 자기자신에로의 지시자(《.》파일에로의 지시자)와 자기어미등록부에로의 지시자(《..》파일에로의 지시자)를 포함한다. 매 파일은 소유자를 가지는데 보통 소유자는 파일을 창조한 사용자이다. 매 파일은 어느 한 그룹에 속한다. Unix판본에 따라 새롭게 창조된 파일은 그의 창조자의 그룹이든가 아니면 그의 등록부의 그룹에 속한다. 이노드안의 마당들을 론하기전에 지령 ls -l로 등록부를 조사하여 다음의 목록을 얻는다.

```
-rw-r--r--1 dieter staff 1617 oct 28 11:o1 adcryp.tex
drwx------2 dieter staff 512 oct 25 17:44 ads/
```

여기에는 다음과 같은 정보가 포함되여 있다.

mode	파일형과 접근권한
uid	파일을 소유한 사용자
gid	파일을 소유한 그룹
atime	접근시간
mtime	변경시간
itime	이노드변경
block count	파일크기
	물리적위치

그림 6-5. 이노드안의 선택된 마당들

- 첫 문자는 파일의 형태를 준다.《-》는 한개의 파일, 《d》는 등록부,《b》는 블로크형장치파일, 《c》는 문자형장치파일을 나타낸다.
- 다음 9개 문자들은 아래에서 론하는 허가(permission)를 나타낸다.

- 그뒤의 수자마당은 파일에로의 련결수를 세는 련결계수기(link counter) 마당이다.
- 그다음 두개의 마당들은 파일의 소유자와 그룹의 이름마당들이다.
- 그다음은 바이트단위의 파일크기이다.
- 시간과 날자는 마지막으로 수정한 시간인 mtime이다. **ls-lu**는 마지막접근시간인 atime을 현시한다. **ls-lc**는 이노드를 마지막으로 수정한 시간인 itime을 현시한다.
- 마지막기입항목은 파일의 이름이다. **ads** 다음의 《/》기호는 등록부를 의미한다.

파일허가(허가비트들)는 소유자, 그룹, 기타를 위한 읽기, 쓰기, 실행접근을 정의하는 9개의 그룹들로 각각 묶어 진다. 《-》는 권한이 없다는것을 나타낸다. 따라서 rw-r--r--는 소유자에게 읽기와 쓰기접근을 주고 그룹와 기타에는 읽기접근을 주며 rwx------은 소유자에게는 읽기, 쓰기, 실행접근을 주고 그룹와 기타에는 아무런 권한도 주지 않는다.

ls -l이 SUID프로그람을 현시하면 여기서는 소유자의 실행허가가 x대신에 s로 주어진다.

-rws—x--3 root bin 16384 Nov 16 1996 passwd*

ls -l이 SGID프로그람을 현시하면 여기서는 그룹의 실행허가가 x대신에 s로 주어진다.

허가비트들은 다음의 순서로 검사된다.

- 만일 **uid**에 파일의 소유자로 되여 있으면 소유자에 대한 허가비트들은 사용자가 그 파일에 접근할수 있는가 없는가를 결정한다.
- 만일 파일의 소유자는 아니지만 **gid**에 그룹이 파일을 소유한것으로 되여 있으면 그룹에 대한 허가비트들은 사용자가 그 파일에 접근할수 있는가 없는가를 결정한다.
- 만일 파일의 소유자도 아니고 파일을 소유한 그룹이나 성원도 아니면 기타에 대한 허가비트들이 사용자가 그 파일에 접근할수 있는가 없는가를 결정한다.

허가비트들을 설정함으로써 파일소유자가 다른 사용자들보다 더 적게 접근하도록 할수 있다. 뜻밖일수 있지만 이 사실은 또한 하나의 일반적인 귀중한 교훈으로 된다. 어떤 접근조종기구에 대하여 서로 다른 접근기준이 어떤 순서로 검사되는가를 명백히 알아야 한다.

2. 허가의 변경

파일의 허가비트는 파일의 소유자 또는 특권사용자만이 실행할수 있는 **chmod** 지령으로 변화시킬수 있다. 이 지령형식은 다음과 같다.

chmod[-fR] absilute file　　모든 허가비트들에 값들을 준다.
chmod[-fR] [who]+pernisson file　허가를 추가한다.
chmod[-fR] [who]-pernisson file　허가를 제거한다.
chmod[-fR] [who]=pernisson file　지적한대로 허가를 재설정한다.

절대방식에서는 파일허가를 8진수로 직접 서술한다. 이 수들은 그림 6-6 으로부터 결정된다. 권한들의 조합은 대응하는 수자들의 합이다. 실례로 허가 rw-r--r--는 지령chmod 644 로 설정한다. 지령 chmod 777은 소유자, 그룹, 기타에게 모든 권한을 준다.

4000	사용자ID를 실행으로 설정
2000	그룹ID를 실행으로 설정
1000	고정비트를 설정
0400	소유자에 의한 읽기
0200	소유자에 의한 쓰기
0100	소유자에 의한 실행
0040	그룹에 의한 읽기
0020	그룹에 의한 쓰기
0010	그룹에 의한 실행
0004	기타에 의한 읽기
0002	기타에 의한 쓰기
0001	기타에 의한 실행

그림 6-6. 접근허가의 8진수표현

기호방식에서는 현재파일허가들이 변경된다. who 파라메터는 다음의 값들을 가질수 있다.

u 소유자허가를 변경시킨다.
g 그룹허가를 변경시킨다.
o 기타허가를 변경시킨다.
a 모든 허가를 변경시킨다.

허가파라메터는 다음의 값들을 취할수 있다.

r 읽기허가
w 쓰기허가
x 파일에 대한 실행허가, 등록부에 대한 탐색허가
X 만일 파일이 등록부이거나 적어도 한개의 실행비트가 설정되였을 때에만 실행허가
s 사용자ID설정허가 또는 그룹ID설정허가

t 본문보존허가(고정비트설정)

-f 선택항목은 오유통보문을 표시하고 -**R** 선택항목은 현재등록부의 모든 부분등록부들에 지정된 변경들을 재귀적으로 적용한다. 프로그람의 SUID허가는 다음과 같이 설정할수 있다.

chmod 4555 file	SUID 기발설정
chmod u+s file	SUID 기발설정
chmod 555 file	SUID 기발지우기
chmod u-s file	SUID 기발지우기

GUID허가는 u 선택항목대신에 g를 리용하여 설정할수 있다.

chown 지령은 파일의 소유자를 변경시키며 chgrp지령은 파일의 그룹을 변경시킨다. chown지령은 환영할수 없는 SUID프로그람의 잠재적인 원천으로 될수 있다. 사용자는 SUID프로그람을 만들어 놓은 다음 소유자를 뿌리로 변경시킬수 있다. 이러한 공격을 방지하기 위해 특권사용자만이 chown 지령을 실행하도록 하고 있다. 어떤 판본들은 사용자가 자기의 파일들에 chown 지령을 적용하게 하며 SUID 와 SGID비트들을 제거한 chown을 가진다. chgrp 지령도 이와 류사하다.

3. 기정허가

편집기나 콤파일러와 같은 Unix 편의프로그람(utility)들은 표준적으로 새로운 파일을 창조할 때에는 기정허가 666을 리용하며 새로운 프로그람을 창조할 때에는 기정허가 777을 리용한다. 이 허가들은 umask 지령에 의하여 후에 조절할수 있다. umask는 금지해야 할 권한들을 렬거하는 3자리 8진수이다. 따라서 umask 777 모든 접근을 금지하며 umask 000은 아무런 제한도 주지 않는다. 실용적인 기정설정값들은 다음과 같다.

022 소유자에 대한 모든 허가, 그룹과 기타에 대한 읽기와 실행허가;
037 소유자에 대한 모든 허가, 그룹의 읽기허가, 기타에 대한 허가금지;
077 소유자에 대한 모든 허가, 그룹와 기타에 대한 허가금지.

실제의 기정허가는 Unix 봉사프로그람의 기정허가를 umask로 마스크하여 끌어 낼수 있다. 즉 기정허가의 비트들과 umask의 반전비트의 론리곱하기를 진행한다. 실례로 기정허가 666과 umask 077은

```
             0 6 6 6
  AND NOT   (0 0 7 7)
  ---------------------
             0 6 0 0
```

로 되여 파일소유자에게 읽기와 쓰기접근이 허락되며 다른 모든 접근은 금지된다. umask는

umask[-S][mask]

지령으로 변화시킬수 있다. 여기서 기발 -S는 기호방식을 의미한다. 마스크가 지적되지 않으면 현재의 umask가 현시된다.

/etc/profile 안의 umask는 체계전반의 기정설정을 정의한다. 이 기정설정들은 특별한 Unix설치방법에 의존하는데 /etc/profile, profile, login, 혹은 .cshrc 와 같은 파일안에 있는 사용자의 홈등록부안에 umask를 설정함으로써 개별적사용자에 대하여 취소할수 있다. VMS와 같은 다른 조작체계와는 달리 등록부들에 대한 개별적인 기정허가들을 정의할수 없으며 파일들이 다른 등록부로부터 자기의 허가를 계승하도록 하였다.

복사지령 cp를 리용하여 새로운 파일을 만들 때 그 파일의 허가는 umask로부터 유도된다. mv지령을 리용하여 현존하는 파일의 이름을 변경하여 새로운 파일을 만들 때에는 현존하는 허가를 그대로 유지한다.

4. 등록부에 대한 허가

매 사용자는 /home/staff/dieter 와 같은 홈등록부를 가진다. 부분등록부는 mkdir 지령으로 만든다. 사용자가 등록부안에 파일들과 부분등록부들을 배치하자면 그 등록부에 대한 정확한 파일허가를 가져야 한다.

- 읽기허가는 ls 또는 그 비슷한 지령들을 실행하여 등록부안에 어떤 파일들이 있는가를 알아 내게 한다.
- 쓰기허가는 등록부로부터 파일을 추가하거나 삭제하도록 한다.
- 실행허가는 현재등록부에 등록부를 만들고 등록부안의 파일을 열게 한다. 만일 파일이 존재한다는것을 알면 그 등록부안에서 그 파일을 열수 있지만 그 등록부안에 무엇이 있는가를 보기 위하여 ls 명령을 사용할수는 없다.

따라서 자기가 소유한 파일에 접근하려면 등록부안에서의 실행허가가 필요하게 된다. 다른 사용자들이 자기의 파일들을 읽어 보는것을 막자면 접근허가를 적절히 설정하든가 또는 등록부에로의 접근을 막아야 한다. 파일을 지우자면 등록부에 대한 쓰기접근과 실행접근이 필요할뿐 파일 그자체에 대한 어떤 허가도 요구되지 않는다. 파일은 지어 다른 사용자의것이여도 무방하다. 이 특징에 관한 체계관리자의 의견을 인용하면

> 만일 당신이 노력하여 누군가의 등록부안에 영구적인 파일을 설치하려면 고통스러울것이다.

Unix의 초기판본들의 유물은 고정비트이다. 이 비트는 프로그람의 본문토막이 처음으로 리용된 다음 가상기억기안에 남아 있도록 한다. 따라서 체계는 페지화구역안으로 자주 접근되는 프로그람들의 코드전송을 피하게 하였다. OSF/1에서는 이 허가를 지정할수 있으나 효과가 없다. 고정비트가 설정된 상태에서 **ls-l지령으로** 파일을 현시하면 x대신 t가 기타에 대한 실행허가로 나타난다. 만일 어떤 등록부의 고정비트를 설정하면 그안에서의 지우기가 제한된다. 고정(sticky)등록부안의 항목은 사용자가 그

등록부에 대한 쓰기허가를 가지며 또 사용자가 해당 파일이나 그 등록부의 소유자이거나 또는 특권사용자인 경우에만 지우거나 이름변경을 할수 있다.

제5절. 일반적인 보안원리들의 실례

이 절에서는 앞에서 본 일반적인 보안원리들이 실천에서 어떻게 쓰이는가를 Unix를 통하여 보기로 한다.

1. 통제된 호출

통과암호라든가 구성파일과 같이 많은 사용자들이 접근할수 있는 민감한 자원들은 소유권, 허가비트, SUID프로그람의 개념들을 결합한 통제된 호출(Controlled Invocation)기구를 리용하여 보호할수 있다.

- 자원과 그 자원에 접근할 필요가 있는 모든 프로그람들을 소유한 새로운 UID를 창조한다.
- 자원에 대하여서는 반드시 그의 소유자에게만 접근허가를 준다.
- 자원에 접근하는 모든 프로그람들을 SUID프로그람으로 정의한다.

지나친 보호를 주의할것. 만일 어떤 파일에 대한 사용자의 직접접근을 금지시켰다면 SUID 프로그람을 통한 간접접근을 제공해 주어야 한다. 원만치 못한 SUID 프로그람은 명백하게 지적된 허가비트들보다 사용자에게 더 많은 직접접근기회를 조성해 준다. 이것은 자원과 SUID 프로그람의 소유자가 뿌리와 같은 특권사용자인 경우에 성립된다.

여기서는 또한 안전체계설계에서 자주 제기되는 처리의 실례들도 취급한다. 체계에서는 자료구조에 의하여 추상적인 속성을 표현한다. 이 자료구조는 여러가지 목적으로 다른 보안기구들에서도 리용된다. UID는 체계안에서 실제적인 사용자를 나타내기 위하여 도입되였다. 그러나 지금 UID는 실제적인 사용자가 아니라 새로운 종류의 접근조종을 위하여 리용된다. 제6장 6절 1에서는 이러한 설계결심들에서 발생할수 있는 문제들을 보여 준다.

2. 파일지우기

흥미 있는 화제번호 2: 론리적 및 물리적기억구조.

만일 파일체계에서 어떤 파일을 지운다면 어떻게 되겠는가? 그 파일이 여전히 어떤 형태로 남아 있겠는가?

Unix에는 파일을 복사하는 두가지 방법이 있다. 내용은 동일하나 독립적으로 존재하는 복사본을 창조하는 **cp**지령과 함께 원본파일에로의 지시자를 가지는 새로운 파일이름을 만들고 원본파일의 련결계수기(link counter)를 증가시키는 (**link**, **ln**)지령이 있다. 둘째 방법에서 새 파일은 원본파일과 내용을 공유한다. 만일 원본파일이 **rm** 또는 **rmdir**

에 의하여 지워 지면 그 파일은 어미등록부에서 나타나지는 않지만 그의 복사본은 물론 내용도 여전히 존재한다. 사용자는 파일들이 여전히 다른 등록부에 존재하고 자기가 그것을 소유하고 있음에도 불구하고 파일을 지웠다고 생각할수 있다. 만일 어떤 파일을 지웠다고 담보하려면 특권사용자는 **ncheck**를 실행시켜 그 파일에로의 모든 련결들을 현시한 다음 그 련결들을 모두 지워야 한다. 한편 다른 처리가 그 파일을 열어 놓은 경우에는 그의 소유자가 제거한다 해도 그 파일은 그 처리가 파일을 닫을 때까지 존재해 있게 된다.

일단 파일이 기억공간에서 지워 졌다면 그 기억공간은 다시 리용할수 있다. 그러나 이 기억장소가 실제로 다시 리용될 때까지 여전히 파일내용을 포함하고 있다. 이러한 잔류기억을 피하자면 파일을 지우기전에 파일의 내용을 모두 0으로 지우든가 또는 기억매체에 적당한 다른 내용을 써넣어야 한다.

3. 장치의 보호

다음의 화제는 여전히 론리기억구조와 물리기억구조사이의 차이에 관계된다. Unix는 장치를 파일처럼 취급한다. 따라서 기억기에 대한 접근이나 인쇄기에 대한 접근은 허가비트의 설정을 통하여 파일에 대한 접근과 비슷하게 조종할수 있다. 장치들은 뿌리에 의해서만 실행할수 있는 **mknod**지령을 리용하여 만든다. **/dev** 등록부안에서 공통적으로 보게 되는 장치들의 작은 표본은 다음과 같다.

/dev/console	조종탁말단
/dev/mem	주기억사영장치(물리기억기의 영상)
/dev/kmem	핵심부기억사영장치(가상기억기의 영상)
/dev/tty	말단

공격자가 파일들을 포함하고 있는 기억장치에 접근할수 있다면 파일과 등록부에 설정된 조종을 우회할수 있다. 만일 기억기에 대한 읽기 또는 쓰기허가비트들이 전체에 대해서 설정되면 공격자는 이 기억기에 보관된 파일에 정의된 허가의 영향을 받지 않고 기억기를 열람하거나 자료를 수정할수 있다. 때문에 거의 모든 장치들은 기타에 의하여 읽을수도 쓰기할수도 없게 되여야 한다.

처리상태지령 **ps**와 같은 지령들은 기억기리용에 대한 정보를 현시하며 따라서 기억기장치에 대한 접근허가를 요구하게 된다. 뿌리프로그람에로의 SUID로서 **ps**를 정의하면 **ps**가 필요한 허가를 얻게 하지만 **ps**지령의 손상은 공격자가 뿌리특권을 가지게 한다. 보다 더 좋은 해결책은 SGID프로그람으로서 **ps**를 취급하며 그룹 **mem**이 기억기장치를 소유하게 하는것이다.

tty말단장치는 또 하나의 흥미 있는 실례이다. 사용자가 가입할 때 말단파일은 사용자에게 할당되며 작업이 끝날 때까지 사용자가 그 파일의 소유자로 된다(말단파일이 리용되지 않으면 뿌리가 소유한다). 이 파일을 기타가 읽을수도 쓸수도 있게 만들어서 사용자가 기타부분으로부터 통보를 받도록 하면 편리하다. 그러나 이것은 결함이 있다. 기타부분들이 사용자의 통과암호를 잠재적으로 포함하고 있는, 말단으로 들어 오고 나가는 모든 통신량을 감시할수 있다. 그들은 실례로 기능건을 재프로그람하여 사용자의 말단에로 지령 등을 보낼수 있으며 이러한 지령들은 사용자가 모르게 실행될수 있다. 일부체계들에서 지능말단들은 일부 지령들을 자동적으로 수행한다. 이것은 공격자에게 다른 사용자의 특권을 리용하는 지령들을 제출할수 있는 기회를 준다.

4. 파일체계의 설치

만일 체계에 여러가지 보안령역이 있고 어느 한 령역이 다른 령역으로부터 객체를 받아 들일 때에는 그 객체의 접근조종속성을 재정의해야 한다.

Unix파일체계는 《/》로 표시되는 하나의 뿌리아래에 서로 다른 물리적인 장치들에 보관된 파일체계들을 련결하여 구축된다. 이것은 mount지령으로 진행한다. 망이 설치된 환경에서 원격파일체계는 망의 다른 마디로부터 설치할수 있다. 류사하게 사용자들은 자기의 유연성자기원판으로부터 파일체계를 설치할수도 있다.

만일 보안전문가라면 경고를 울리기 시작할것이다. 설치된 파일체계는 례하면 공격자의 등록부에 들어 있는 뿌리프로그람들에로의 SUID와 같은 모든 종류의 불필요한 파일들을 포함할수 있다. 일단 파일체계가 설치되면 공격자는 이 프로그람을 실행시켜 특권사용자의 상태를 얻을수 있게 된다. 기억기에 대한 직접접근이 허락된 장치파일에서도 위험이 생길수 있다. 여기서 허가들은 공격자가 이 파일들에 접근할수 있게 설정되여 있다. 때문에 지령

mount[-r][-o options] 장치등록부

에서 -r기발은 읽기전용설치를 지시하며 선택항목은 다음과 같다.

nosuid : 설치된 파일체계에서 SUID 와 SGID비트들을 제거한다.
noexec : 설치된 파일체계에서 2진파일들은 실행할수 없다.
nodev : 블로크 또는 문자전용장치들은 파일체계로부터 접근할수 없다.

이외에 Unix의 여러 판본들은 mount에 대한 여러가지 선택항목들을 실현한다.

5. 파일체계의 뿌리를 변경

접근조종은 권한 없는 사용자가 접근할수 없는 위치에 객체를 배치함으로써 실현할수 있다. Unix에서 뿌리변경지령 chroot는 권한 없는 사용자가 리용할수 있는 파일체계의 부분을 제한한다. 이 지령은 뿌리만이 실행시킬수 있다.

chroot 등록부지령

이 지령이 실행되면 뿌리등록부를 《/》로부터 《등록부》로 변경시킨다. 그때부터 새로운 뿌리아래 있는 파일들만이 접근가능하다. 만일 이러한 전략을 리용한다면 사용자프로그람이 필요한 모든 체계파일들을 찾아 내는가를 확인해야 한다. 이 파일들은 /bin, /dev, /etc, /tmp, /usr 같은 등록부들에 있을것으로 《예견된다》. 같은 이름을 가진 새로운 등록부들을 새로운 뿌리밑에 만들어 놓고 거기에 사용자가 요구할 파일들을 원래의 등록부로부터 복사하거나 해당한 원본파일에로의 련결을 지어 주어야 한다.

6. 탐색경로

마지막으로 흥미 있는 문제는 《잘못된》 위치에서 취한 프로그람의 실행이다. Unix사용자들은 쉘(지령해석기)을 통하여 조작체계와 호상작용한다. 사용자는 파일체계내에서 프로그람의 위치를 지적하는 경로이름을 완전히 지정하지 않고 프로그람의 이름만을 입력하여 간단히 프로그람을 실행시킬수 있다. 이때 쉘은 사용자의 홈등록부안의 .profile에 주어 지는 PATH환경변수에 지정된 탐색경로를 따라 가면서 프로그람을

찾는다(홈등록부안의 모든 파일들을 보려면 ls-a를 리용하고 자기의 .profile을 보려면 more.profile을 리용한다). 지적된 이름과 같은 프로그람을 포함하는 등록부가 발견되면 탐색을 정지하고 그 프로그람을 실행한다. 표준적인 탐색경로는 다음과 같다. 실례:

PATH=.:$HOME/bin:/usr/ucb:/bin:/usr/bin:/usr/local:/usr/new:/usr/hosts

이 실례에서 탐색경로안의 등록부들은 《:》로 구별한다. 첫 항목 《.》는 현재등록부이다. 현존하는 어떤 프로그람과 같은 이름을 달고 원본프로그람이 있는 등록부보다 먼저 탐색되는 등록부에 배치하는 방법으로 트로이목마를 침입시킬수 있다.

이러한 공격을 막자면 프로그람을 호출할 때 그의 완전한 경로 실례로 su대신에 bin/su로 지정하여야 한다. 같은 문제의 또 다른 표현은 동반비루스의 침입이다(제8장 8절 5).

제6절. 검열기록과 침입검출

일단 체계가 설치되고 동작할수 있게 되면 그의 보안기구는 위법사용자의 작용을 막아야 한다. 그러나 보안기구가 적당치 못하거나 결함이 있을수 있다. Unix와 같이 복잡한 체계에서는 바람직하지 않은 보안설정이 체계실행을 위하여 필수적일수도 있다. 때문에 보안위반의 발생이나 보안위반의 발생후에 보안침해나 다른 수상한 사건이 일어 나는가를 검출하는 추가적인 기구를 가지는것이 좋다. 이러한 기구들을 부류별로 가르면 다음과 같다.

- 검열(audit): 보안관계사건들을 후에 분석하기 위해 검열기록(검열궤적)에 기록한다.
- 침입검출 : 이상스러운 사건이 발생하면 그것을 검출하고 전자우편이나 조종탁통보문으로 체계관리자에게 알려 준다.
- 자동보복(침입응답): 보안경보에 적절한 작용으로 즉시에 반작용한다. 실례로 수상한 사용자는 체계로부터 자동적으로 추방된다. 그러나 거짓경고가 있을수 있고 또한 자동보복이 항상 좋은 결과를 줄것인가는 명백치 않다. 안전림계체계에서는 침입검출이 권고될수 있으나 사용자들이 침입검출체계에 걸려 들수 있는 례외적인 행위를 해야 하는것은 엄밀히 비상사태의 때이다.

검열기록을 안전한 장소에 보관하는것이 중요하다. 검열기록을 변경시킬수 있는 공격자는 자기의 흔적을 완전무결하게 숨길수 있다. 다음목록은 안전도가 증가하는 순서로 검열기록을 보호하기 위한 선택항목을 준다.

1. 검열기록에 론리적인 보호를 설정하여 오직 특권사용자만이 쓰기접근을 할수 있게 한다.
2. 검열된 기계의 뿌리가 특권사용자의 특권을 가지지 못할 때에는 검열기록을 다른 콤퓨터에로 보낸다. 이 방법은 2중보호를 제공한다. 첫째로 합법적인 특권사용자의 작용보다 더 우월한 조종이다. 둘째로 검열된 기계의 뿌리접근을 얻으려는 공격자가 자기의 흔적을 감추려면 또 다른 콤퓨터에 침입해야 한다.

3. 검열기록을 콤파일러나 편집기,일정한 망봉사프로그람들과 같은 불필요한 봉사프로그람들이 제거된 전용검열장치에로 보낸다. 그러면 공격자는 그 검열장치에로의 뿌리접근을 하기가 훨씬 더 어렵게 된다.
4. 검열기록을 안전한 인쇄기에로 보낸다. 다음은 물리적인 보안대책들이 검열기록의 완정성을 보호하게 된다.

관리자적인 립장에서 기록해야 할 보안관련사건들과 검열기록을 보관하는 시간을 결정하여야 한다. 물론 기록된 여러가지 사건들의 수와 검열기록을 보는 조작자의 능력사이에는 이률배반관계가 있다. 보안과 관련되는 사건들이 많을수록 검열기록은 더욱더 확장되며 침입시도들에 대한 단서를 잡기 어렵게 된다. 다른 한편 만일 기록되는 사건들이 너무 적으면 공격이 일단 검출되였을 때 공격이 어떻게 수행되였는지 확증하기가 어렵게 된다. 게다가 사용자의 행위를 기록하는것자체가 사회적으로 일정하게 제한된다. 즉 그것은 국가적인 비밀법 또는 리용규칙의 일부분으로 될수 있다. 일부 보안관계사건들은 Unix log 파일안에 다음과 같이 자동적으로 기록된다.

/usr/adm/latlog : 사용자가 마지막으로 가입한 시간을 기록. 이 정보는 finger 지령으로 볼수 있다.
/var/adm/ntmp : who지령에 의하여 리용된 등록자리정보를 기록.
/var/adm/vtmp : 사용자가 가입 혹은 탈퇴할 때에 매번 시간을 기록. 이 정보는 last지령으로 현시할수 있다. 이 파일이 모두에게 리용가능한 기억기에 넘어 지는것을 막기 위해서 규칙적인 간격마다 자동적으로 없애 버릴수 있다.
/var/aadm/acct : 실행된 모든 지령을 기록. lastcomm지령으로 현시할수 있다.

이 파일들의 정확한 이름과 위치는 Unix체계마다 다르다. accton지령에 의하여 펼쳐 진 등록자리는 검열목적에도 리용할수 있다. 이밖에 Unix체계를 관찰하는 지령들로서는 find, grep, ps, users 들이 더 있다. 기술적측면에서 검열기록이 그에 할당된 기억공간을 초과할 때에는 어떻게 하겠는가를 결정해야 한다.

- 검열기록을 처음부터 다시 써야 하는가?
- 검열기록을 이 목적으로 할당되지 않은 공간안에 써야 하는가?
- 체계는 관리를 위한 작용이 진행되는 동안 정지하고 대기해야 하는가?

검열기록은 길고 지루하므로 이러한 검열기록을 검열하는 조작자가 중요한 사건을 놓칠수 있는 가능성이 많다. 때문에 보안위반을 나타내는 지시기를 찾아 보든가 수상한 행동을 식별해 내기 위하여 인공지능적인 방법으로 검열기록을 조사하는 전문가체계를 리용하자는 의견들이 제기되고 있다. 침입검출체계들에서도 비슷한 방법을 리용한다.

1. 가입사용자식별자

앞에서 본 기록파일안에 기록된 보안관계자료목록을 보자. 이 사건들중 대부분이 사용자와 관련되므로 기록항목은 사건을 발생시킨 처리의 UID를 포함해야 한다. 그다음 검열은 SUID프로그람에 의하여 어떤 영향을 받겠는가? 이러한 프로그람은 그 소유자의 UID로 실행하지만 그 프로그람을 실행시키는 사용자의 UID로는 실행되지 않는다. 여기

서 기록항목들은 기록을 위한 현행처리의 UID를 리용할 때 실지로 필요한 성분들을 제공해 주지 못한다.

책임추적가능성을 높이기 위해 Unix의 안전한 판본들은 최초에 가입한 사용자의 신원정보를 보관한다. 실례로 SCO Unix는 이 목적으로 가입사용자식별자 (LUID)를 리용한다(HP-UX 에서는 효과적인 사용자신원이라는 개념을 사용한다). LUID는 가입시에 만들어 진다. 자식처리는 어미로부터 LUID를 계승한다.

사용자신원들은 두가지 목적 즉 접근조종과 책임추적가능성에 쓰이는 보안속성이다. 같은 시각에 두가지 목적에 꼭 같은 속성을 리용하는것이 언제나 가능한것은 아니다. UID가 《실지》사용자에 대응하는 한 허가에 기초한 접근조종과 검열은 호상 보충적으로 작용한다. 일단 SUID나 SGID프로그람을 통한 자원접근을 막기 위하여 특수한 사용자신원을 만든다면 검열시 리용에서 제한을 받게 된다.

제7절. 포장기

지금까지 본 접근조종과 검열기구들은 그리 세련된것들은 아니다. 그것들은 조작체계보안의 전통에 따라 자원에 대한 접근조종에 초점을 두고 있다. 기초적인 접근조종기구들을 잘 리용하면 《중간준위》에서 조종을 실현할수 있다. 대신에 이 목적을 위하여 Unix자체를 수정할수도 있다. 여기서의 과제는 조작체계의 다른 요소들에는 영향을 주지 않으면서 유용한 보안조종들을 추가하는 방법으로 변경시킬수 있는 Unix요소를 찾아내는것이다. Unix가 복잡한 체계이므로 간단한 과제가 아니다.

TCP포장기들은 아주 재치 있게 이 설계수법을 례증한다. **telnet**나 **ftp**같은 Unix 망봉사들은 다음의 원리에 기초하여 구축된다. **inetd** 데몬은 들어 오는 망접속들을 주시한다. 접속이 이루어 지면 **inetd**는 적절한 봉사기프로그람을 기동시킨후 다음접속을 대기하는 상태로 돌아 간다. 이 데몬은 많은 봉사프로그람들의 작업을 조종하기때문에 고급봉사기(super service)라고 한다. **inetd** 데몬은 봉사들(포구번호들)을 프로그람들에로 넘기는 구성파일을 가지고 있다. 이 구성파일의 기입항목은 다음의 형식을 가진다.

Service type protocol waitflag userid executable command-line

실례로 **telnet**기입항목은 다음과 같다.

telnet stream tcp nowait root/usr/bin/in.telnetd in.telnet

inetd는 자기가 조종하는 봉사에 대한 요구를 받으면 구성파일을 참고하여 지정된 실행가능한 처리(executable)를 실행하는 새로운 처리를 만든다. 이 새로운 처리의 이름은 지령행(command-line)마당에 주어 진 이름으로 변경된다.

보통 실행가능한 처리의 이름과 지령행에 주어 진 이름은 같다. 이것이 훌륭한 계교의 실머리를 준다. 본래의 실행가능한 프로그람대신 포장기프로그람에로의 **inetd** 데몬을

지적해 주고 포장기가 자기의 보안조종을 수행한 다음 실행하려는 본래의 실행가능한 프로그람의 이름을 되살리기 위하여 처리의 이름을 리용한다. 이 실례에서 telnet에 대한 구성파일기입항목은 다음과 같이 바꿀수 있다.

telnet stream tcp nowait root/usr/bin/tcpd in.telnet

지금 실행된 프로그람은 /usr/bin/tcpd이다. 이것은 실행가능한 TCP포장기이다. 포장기를 실행하는 처리는 여전히 in.telnet라고 부른다. 이 포장기안에서 모든 접근조종을 수행할수 있거나 또는 원하는것을 기록할수 있다. 본래의 응용프로그람에서 포장기는 IP주소의 려파(filtering)에 리용되였다(제13장). 포장기는 자기가 들어 있는 등록부례하면 /usr/bin과 자기자체의 이름 즉 in.telnet를 알기때문에 그다음에 본래의 봉사기프로그람 즉 /usr/bin/in.telnet을 호출한다. 사용자는 이 차이를 느끼지 못하며 결국 이전과 꼭 같은 봉사를 받게 된다.

다른 하나의 간접준위를 추가하는것은 콤퓨터과학의 위력한 도구이다. 보안에서 이 도구는 체계를 공격하는데와 체계를 보호하는데 리용할수 있다. Inetd 데몬과 봉사기프로그람사이에 TCP 포장기를 끼워 넣으면 데몬의 원천코드나 봉사기프로그람의 원천고드를 변경시키지 않고도 보안조종을 추가할수 있게 된다.

이 실례의 매력은 그의 일반성에 있다. Unix망봉사기의 전반체계를 보호하는데도 꼭 같은 원리를 리용할수 있다.

TCP 포장기는 기본설계원리 즉 통제된 호출(controlled invocation)과 봉사를 호출하는 프로그람을 변경시키지 않고 그 봉사에 대한 보안검사를 추가할수 있게 하는 세련된 기교(elegant trick)를 결합하고 있다. 이것은 현존하는 체계에 보안을 새로 구축해야 할 때에도 도입할수 있는 원리이다.

제8절. 설치와 구성

조작체계에서 가장 중요한 점은 그의 설치이다. 조작체계는 많은 보안특징들과 보안에 영향을 주는 특징들을 가지고 있다. 이 특징들중의 일부는 문서화가 잘되여 있지 않다. 기정설정은 순조로운 설치와 조작에 유리하다. 이 설정들은 운영기술자나 체계관리자들에게 많은 특권을 준다. 모든 다른 사용자들과 같이 체계관리자를 제한하고 체계관리자들과 보안관리자들을 분리시키는것이 좋다. 복잡하고 어렵게 문서화된 특징들은 계획된 보안방책을 효과적으로 시행하도록 체계를 설치하기 어렵게 한다. Unix는 체계관리자의 일감을 돕는다.

체계관리자는 모든 보안관계파일들에 대하여 해박한 지식을 가져야 하며 설치한 다음 변화시켜야 할 위험한 기정설정들에 대해서도 잘 알아야 한다.

체계가 설치되면 보안관련정수들을 표준 Unix편집지령으로 정의한다. 자원에 대한 허가는 응용프로그람보다 조작체계쪽에 더 가까운 준위로 설정한다.실례로

/etc/passwd 같은 파일들을 편집하여 사용자를 설정한다. passwd프로그람의 보호는 다음의 지령의 영향을 받는다.

chmod 4750 /bin/passwd chgrp staff/bin/passwd

체계를 검열할 때 Unix는 리용되는 지령들을 조사한다. 실례로 다음의 지령은 통과암호없이 등록자리들을 조사한다.

awk-F: ‘length($2)<1 {print $1}’ </etc/passwd

SUID와 SGID는 다음지령으로 찾을수 있다.

find/-type f\ (-perm 2000 -o -perm 4000\ ) -exec 1s -1d}}\ ;

접근조종방책은 단순한 자유접근조종을 통하여 지원된다. 구조화된 보호는 그룹성원자격에 준하여 그리고 가입이 금지된 등록자리를 리용하여 실현할수 있다.

이와 같이 보안특징들을 관리하며 현재보안상태를 검사하기 위한 추가형 Unix보안제품들이 있다. 널리 쓰이는 검사도구들은 COPS[54]([50]도 역시)와 SATAN이다. 이것들은 약한 통과암호, 파일이나 등록부에 대한 잘못된 허가 그리고 잘못된 구성파일과 같은 결함들을 탐색한다. 체계관리자가 자기의 체계안에서 쉽게 공격 받을수 있는 약점을 검출하기 위하여 이 도구들을 리용할수 있지만 같은 목적을 가진 공격자도 역시 리용할수 있는 방법이기때문에 광범히 리용되지 못한다.

이 장의 문헌안내

이 장에서는 Unix 보안의 일부를 취급하였다. 참고문헌 [36, 50, 55, 162]들에 보다 많은 내용을 준다. Unix보안에 관한 SRI기술보고서(참고문헌[35])는 다음의 Web페지에 있다.

http://www.sri.ucl.ac.be/SRI/dicuments/unic-secure

만일 개별적인 Unix판본들에 대한 고유한 정보가 필요하다면 체계에 부속된 직결문서와 제작자가 제공하는 문서를 구해 보시오. 이 장에서 취급한 대부분의 실례들은 DEC-Unix설치에 관하여 검토되였으므로 각이한 Unix판본들에 대해서는 수정이 요구될수 있다. Linux는 Unix의 무료쏘프트웨어이다. Linux에 관한 보안정보의 원천은 다음의 Web페지에 있다.

http://bach.cis.temple.edu/linux/linux-secarity/

다중보안 Unix 체계들은 참고문헌[132]에서 론의한다. CERT는 보안제품들에서 발견되는 결함들에 대한 정기적인 의견을 출판하고 있다. 일반정보는 Web페지에 있다.

htpp://into.cert.org/pub/cert.adv:sorjes/

웨네마(Wietse Venema)의 Web페지는 다음과 같다.

ftp://ttp.win.tue.m/pub/security/iadex.html

이 페지는 Unix보안에 관계되는 도구와 론문들을 취급한다. SRI에서 침입검출전문가체계에 관한 연구론문이 참고문헌[92, 72]들에 주어 진다. Unix검열제품에 관한 Web페지는 다음과 같다.

http://www.axent.com
http://www.ov.com

침입검출체계에 대한 최근의 분석자료는 다음의 Web페지에서 볼수 있다.

http://www.secnet.com/nav2.html

련습문제

1. 보안관련지령에 대한 직결문서를 조사하시오.
   /etc/passwd안에서 자기의 항목을 찾고 자신의 파일과 등록부에 대한 허가설정을 검사하시오.

2. 홈등록부에 부분등록부를 만들고 이 부분등록부에 짧은 통보문을 담은 welcome.txt파일을 배치하시오. 부분등록부에 허가비트들을 설정하여 소유자가 실행접근을 가지게 하시오.

   - cd로 부분등록부를 현행등록부로 하시오.
   - 부분등록부목록을 보시오.
   - welcome.txt 의 내용을 현시하시오.
   - 부분등록부에 welcome.txt의 복사문을 만들어 놓으시오.
     부분등록부에서 읽기허가와 쓰기허가에 대하여 같은 실행을 반복하시오.

3. 다른 사용자들로부터 tty장치를 어떻게 보호하는가?

4. UID와 GID 그리고 VSTa능력의 골격내에서 허가를 통하여 Unix접근조종을 얻을수 있는가?

5. 보안로출을 줄이기 위해 여벌만들기절차를 어떻게 설정하겠는가?

6. Unix보안기구들로 《장성》보안과 클라크-윌슨의 모형을 실현하시오.

7. 어느 Unix지령이 등록부안의 모든 전체쓰기가능한 파일들을 현시하는가?

8. Unix보안에서 강한 점과 약한 점을 지적하시오. 이 문제로 소론문을 쓰시오(1000단어).

제7장. Windows NT 보안

앞에서 본 Unix보안은 조작체계가 제공하는 초보적인 보안기구들에 대한 학습이였으며 조직들의 보안방책을 실현하는 문제에 대해서는 거의 다치지 못하였다. 조종의 추가적인 중간층들은 이 문제를 취급하는데 쓰인다. 기타 실천에서 중요한 령역들은 체계관리자들에 의한 보안속성과 보안조종의 관리이다. 이 장에서는 Windows NT보안을 학습하면서 이 방향으로 한걸음 더 전진하게 된다.

목적

- 조작체계가 보안방책실현을 어떻게 지원하는가를 고찰한다.
- 콤퓨터망에 고유한 보안문제들을 조사한다.
- 체계관리와 관련되는 몇가지 주요보안문제들을 제기한다.
- Windows NT보안의 기초개념을 준다.

제1절. 소개

Windows NT는 다양한 처리기구성방식에서 동작할수 있는 이식성 있는 조작체계로서 마이크로쏘프트회사가 개발한 체계이다. Windows NT는 뒤방향으로 호환성을 가진다. 즉 이 체계에서 MS-DOS, OS/2, Windows 응용프로그람들을 실행시킬수 있다. Windows NT는 망련결능력을 포함하여 POSIX순응성을 가지며 단독구성에서 판본3.51은 c2클라스의 인정을 받았다. Windows NT가 보안관리를 지원하는것은 그라프대면부로 보안특징들을 관리할수 있게 해주는것뿐이 아니다.

앞에서와 마찬가지로 이 장에서는 Windows NT보안에 대한 완전한 개괄이나 그의 약점에 대한 구체적인 평가 또는 그의 모든 보안특징들을 최대로 리용하기 위한 안내는 주려 하지 않는다. 이 장의 뒤에 있는 안내도서들에서 그에 대한 구체적인 자료들의 출처를 지적한다. 이 장의 기본목적은 아직 취급하지 않은 콤퓨터보안의 기본문제들을 실천적으로 례증하는 즉 많은 사람들이 흥미를 가지는 특징들을 강조하는것이다. Unix보안과 비교해 볼 때 Windows NT는 사람－기계척도에서 사람쪽에 더 가까운 우월한 측면을 가지고 있다(그림 7-1).

1. 보안구성방식

Windows NT에서 사용한 기초적인 보안기구들을 간단히 보면 다음과 같다.

- Unix에서와 같이 사용자방식(보호고리 3)과 핵심부방식(보호고리 0)의 구별이 있다.

Windows NT의 실행을 포함하여 핵심적인 조작체계봉사프로그람들은 핵심부방식에서 실행된다. 사용자프로그람은 응용프로그람대면(API)을 호출하여 조작체계봉사를 실행시킨다. 상태절환과 고리 3으로부터 0에로의 이행은 국부수속호출기구(Local Procedure Call facility)에 의하여 조종된다.

- 자물쇠잠그기(locking)는 초보적인 조작체계기능이다. 만일 한 사용자가 어떤 객체에 배타적인 자물쇠(lock)를 설정하면 다른 사용자는 접근할수 없게 된다.
- 자료는 소유물형식으로 보관된다. 이러한 형식의 자료들을 처리할수 있는 편의프로그람들은 조종의 중간층으로서 쓰인다. 그러나 이러한 조종들은 조작체계를 변경하지않고도 우회할수 있으며 《애매성에 의한 보안》에 대한 보통의 경고들을 적용한다.
- Windows NT는 객체지향설계이다. 모든것을 《자원》으로 보는 Unix에서의 자유접근조종기구와 대조적으로 Windows NT에서의 자유접근조종은 객체형들에 따라 변할수 있다.
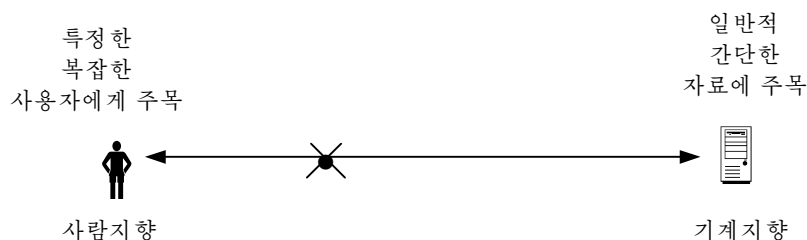
그림 7-1. 사람 - 기계척도에서 Windows NT 보안특징의 위치

조작체계의 다음과 같은 요소들은 보안부분체계의 부분으로 된다.

- **보안참조감시기**(SRM): 접근조종을 책임진다. SRM은 핵심부방식에서 동작하는 실행가능한 요소이다.
- **국부보안권**(LSA): 사용자등록자리를 검사하고 체계접근통표(SAT)를 창조하는 가입시에 포함되는 사용자방식요소이다. LSA는 검사기능책임이 있다.
- **보안등록자리관리자**(SAM): LSA에 의하여 리용된 사용자등록자리자료기지를 유지하며 LSA를 위한 사용자인증을 제공하는 사용자방식구성요소이다.
- **가입처리**: 사용자방식에서 가입할 때 사용자를 인증하는 처리.

제2절. 등록고

등록고(registry)는 Windows NT구성자료를 위한 중심자료기지이다. 등록고의 항목들을 열쇠(암호화에서의 열쇠와 혼돈하지 말아야 한다.)라고 부른다. 등록고는 계층적자료기지이다. 정점준위에서 등록고는 벌집(hive)이라고 부르는 4개의 부분들로 구조화된다. 벌집은 열쇠들(등록부들)을 포함하며 열쇠들은 또 부분열쇠(부분등록

부) 또는 값들(자료항목)을 가진다. 4개의 벌집안에 있는 정점준위열쇠를 흔히 뿌리열쇠라고 부른다. 매 벌집은 검사하려는 벌집 또는 그의 부분열쇠들에로의 변경을 기록하는 기록파일을 하나씩은 가진다(체계벌집은 제외이다. 그의 기록항목은 system.alt파일에 쓴다).

등록고에서는 체계를 사용자의 요구에 맞게 구축할수 있으며 기정보호들이 설정된다. 때문에 등록고자료의 완정성을 보호하는것이 필요하다. 실례로 등록고열쇠는 조작체계가 자동적으로 어떤 실행가능한 파일을 찾는 장소를 가리킬수 있다. Windows NT에서는 이것을 경로(path)라고 부른다.

만일 이러한 열쇠에 대한 허가설정이 무효화되여 누구나 쓰기할수 있으면 공격자는 경로를 수정하여 범죄적인 프로그람을 침투시킬수 있게 된다. 이처럼 높은 준위의 보호와 사용의 편리성은 언제나 동시에 조화되지 않는다.

등록고는 소유물형식으로 보관된다. 등록고를 수정하는 조작체계도구는 오직 등록고편집기뿐이다. 그러나 사용자들은 등록고편집기를 우회하는 자체의 도구를 만들어 등록고열쇠에 직접 접근할수 있다(제7장 3절 3). 체계관리에 리용되지 않는 모든 기계들에서 등록고편집기를 제거하는것은 방어에서 제1선이다. 일부 보안관련열쇠들은 등록고편집기로도 직접 편집할수 없게 해야 하며 오직 체계방책편집기(system policy editor)와 같은 특정의 편의프로그람에서만 변경시킬수 있게 하여야 한다.

1. 등록고파일허가

접근조종목록은 벌집과 열쇠에 대하여 설정된다. 접근허가는 관리자(Administrators), 체계(System), 사용자(Users), 창조자/소유자(Creator/Owner), 모든 사용자(Everyone)와 같은 그룹들에 대하여 정의된다. 가능한 접근허가들은 다음과 같다.

- 읽기전용(read only): 사용자는 읽기만 하고 어떤 변경도 할수 없다.
- 완전조종(full control): 사용자는 편집, 창조, 지우기할수 있으며 또는 열쇠의 소유권을 회수할수 있다.
- 특수조종(special access):사용자는 렬거된 목록에 따라서 허가를 받을수 있다.

특수조종의 표준적인 목록은 다음의 항목들을 포함한다.

- 값묻기(Query Value): 열쇠값을 읽는다.
- 값설정(Set Value): 열쇠값을 설정한다.
- 부분열쇠만들기(Create Subkey): 현존하는 열쇠안에 새로운 부분열쇠를 만든다.
- 열쇠계수(Enumerate Key): 열쇠안의 모든 부분열쇠들을 식별한다.
- 통지(Notify): 열쇠에 의하여 발생된 검열통지를 수신한다.
- 련결창조(Create Link): 열쇠에로의 기호련결을 창조한다.
- 지우기(Delete): 열쇠를 지운다.
- DAC쓰기(Write DAC):열쇠에 대한 접근조종목록을 수정한다.
- 소유자쓰기(Write Owner): 소유권을 획득한다.
- 조종읽기(Read Control): 열쇠안에서 보안정보를 읽는다.

실례로 뿌리열쇠에 대한 권고된 기정파일허가는 다음과 같다.

1. HKEY_LOCAL_MACHINE: 국부콤퓨터, 하드웨어 및 조작체계에 대한 정보를 포함한다.
    - 관리자: 완전조종
    - 체계: 완전조종
    - 모든 사용자: 읽기
2. HKEY_CLASSES_ROOT: 파일련관,OLE(객체의 련결과 매몰) 그리고DDE(동적자료교환)클라스정의에 대한 자료를 담고 있다.
    - 관리자: 완전조종
    - 체계: 완전조종
    - 모든 사용자: 읽기
3. HKEY_CURRENT_USER: 현재 가입한 사용자들에 대한 사용자프로필들을 포함한다.
    - 관리자: 완전조종
    - 체계: 완전조종
    - 사용자: 완전조종
4. HKEY_USERS: 체계에 설치된 모든 사용자에 관한 프로필을 포함한다.
    - 관리자: 완전조종
    - 체계: 완전조종
    - 사용자: 완전조종

HKEY_LOCAL_MACHINE가 가지는 보안관련부분열쇠들은 다음과 같다.

HKEY_LOCAL_MACHINE\ SAM: 사용자와 그룹등록자리의 자료기지
HKEY_LOCAL_MACHINE\ Secure: 보안부분체계에서 리용한 국부적인 보안방책에 대한 정보
HKEY_LOCAL_MACHINE\ Software\ 마이크로쏘프트\ RPC: 원격수속호출
HKEY_LOCAL_MACHINE\ Software\ Microsoft\ Windows NT\ CurrentVersion

이 열쇠들을 변경시킴으로써 공격자는 조작체계의 거동을 변경시킬수 있다.이 열쇠들에 대한 가능한 허가설정은 다음과 같다.

- 관리자: 완전조종
- 체계: 완전조종
- 창조자/소유자: 완전조종
- 모든 사용자: 특수한 접근

특수한 접근은 값묻기,부분열쇠계수, 통지, 조종읽기에 대한 허가를 준다.
만일 어느 한 열쇠가 존재하지 않는다면 어떻게 되겠는가? 이 문제를 례증하기 위해 어떤 사용자나 그룹들이 등록기에 원격으로 접근할수 있는가를 지적하는 열쇠를 고찰하자.

HKEY_LOCAL_MACHINE\ SYSTEM\ CurrentControlSet\ Control\
SecurePipeServers\ Winreg

만일 열쇠가 존재하면 사용자가 그 등록고에 대한 원격접근을 요구할 때 참고될것이며 그 등록고의 원격편집에 특별한 제한을 적용할수 있다. 만일 그러한 열쇠가 존재하지 않는다면 원격접근에 대한 검사가 집행되지 않을것이다. 원격접근은 등록고에 대한 국부접근과 꼭같이 취급된다.

제3절. 식별과 인증

Windows NT는 인증을 위해 사용자이름과 통과암호를 리용한다. 이 기구의 실현은 봉사기들과 워크스테이션들로 구성된 망을 전제로 하고 있다. Windows NT5.0은 인증을 위해 Kerberos를 리용한다.

1. Windows NT통과암호기구

통과암호는 암호화된 형태로 보관된다. 실제로 두개의 암호화된 통과암호 즉 마이크로쏘프트 LAN관리자통과암호와 Windows NT통과암호가 존재한다. 사용자들이 약한 통과암호를 사용하지 못하도록 통과암호려과 동적련결서고(DLL)를 설치할수 있다. 다음의 열쇠가 등록고에 추가되여야 한다.

HKEY_LOCAL_MACHINE\ SYSTEM\ CurrentControlSet\ Control\
LSA\ NotificationPackage\ Passfilt.dll

마이크로쏘프트가 제공하는 통과암호려과기는 통과암호가 적어도 6개 문자를 가지도록 한다. 통과암호는 적어도 다음의 4가지 자료형 즉 대문자, 소문자, 수자, 자모 아닌 기호들중 3가지 형을 포함해야 한다.

통과암호는 사용자이름의 일부분을 포함할수 없다.

암호화된 통과암호는 SAM자료기지안에 있는 사용자등록자리안에 보관된다(제7장 4절 3). SAM자료기지는 등록고의 한부분이다. SAM자료기지는 표준본문편집기로 읽을수 없는 2진파일이다. 암호화된 통과암호를 추출하기 위하여 조작체계는 체계호출이 SAM API를 리용하도록 한다. 조작체계는 이러한 체계호출들이 특권상태에서 실행되는 때에만 허락한다. SAM자료기지는 기정값으로 전체 읽기가능이지만 항상 체계가 리용해야 하므로 잠그어 져 있다. 그러나 체계가 설치되거나 또는 **rdisk**지령이 A:구동기를 출력으로 지정하지 않고 주어 질 때 창조되는 **\ system32\ repair** 등록부안에는 여벌복사본도 존재한다.

중요한 자료의 여벌복사본은 본래의 자료와 동일한 보호를 요구한다.

LAN관리자통과암호

LAN관리자통과암호는 암호화되며 SAM자료기지안에 보관된 암호문으로부터 검색할수 있다. 통과암호는 14개 문자길이로 설정할수 있다. 암호화알고리듬은 128문자까지 허용하지만 통과암호차림표의 입력칸은 14개 문자만을 취한다. 14개 문자길이가 안되는 통과암호들은 나머지를 령으로 채운다. 그다음 대문자로 변환하고 14개 문자(단일바이트의 원본장치제작자(OME)문자모임)들을 두개의 7byte블로크들로 나눈다. 이 블로크들은 기우성을 포함하여 8byte DES로 확장되며 고정마지크열쇠(fixed magic key)들인 0xAA,0xD3,0xB4,0x35,0xB5,0x14,0x4,0xEE 밑에서 암호화된다(실제로 이러한 열쇠에 대하여서는 요술이 없다). 두개의 결과는 련결되여 암호화된 LAN관리자통과암호를 구성한다.

Windows NT통과암호

Windows NT통과암호는 한방향함수를 리용하여 하쉬화된다. Windows NT통과암호는 SAM자료기지안에 보관된 하쉬값으로부터 재생할수 없다. 이 하쉬값을 계산하기 위해 사용자의 통과암호를 먼저 유니코드(65536문자까지 지원하는 16bit문자모임)로 변환하고 MD4알고리듬(제12장 2절 1)을 리용하여 16byte의 하쉬값을 얻는다.

2. 가입

가입은 Windows NT조작체계가입화면을 기동시키는 CTRL+ALT+DEL건들을 누르는것으로 시작한다. 이때 CTRL+ALT+DEL을 안전주의조작렬(secure attention sequence)이라고 부른다. 이 안전주의조작렬은 사용자와 콤퓨터사이의 대화가 시작되였을 때와 지어는 가입화면이 이미 현시되였을 때에도 리용해야 한다. 이것은 응용프로그람에 의해 복제될수 없는 낮은 준위의 Windows NT함수들에 대한 호출을 발생한다. MS-DOS기동디스크로 실행되는 DOS방식프로그람들은 Windows 가입화면을 모방할수 있으며 기만공격을 준비할수 있다[63].

Windows NT는 경고문으로 법률적주의(legal notice)를 현시하는 선택항목을 제공한다. 사용자들은 가입하기에 앞서 이 경고통보문에 응답해야 한다. 그다음 사용자는 사용자이름과 통과암호를 입력한다. 사용자이름과 통과암호들은 가입처리에서 수집되여 국부보안권(LSA)에 넘겨 진다. LSA는 인증패키지를 호출하여 입력된 사용자이름과 통과암호를 등록자리자료기지에 보관된 값들과 비교한다. 일치하면 SAM은 사용자의 보안식별자(SID)와 사용자가 속한 그룹의 보안식별자를 돌려 준다.

인증패키지는 가입대화를 창조하고 모든 SID들과 함께 그 대화를 LSA에 돌려보낸다.

LSA는 사용자의 SID와 사용자권한을 포함하는 체계접근통표(SAT)를 만든다. 그다음 SAT는 가입처리에 의하여 Win32부분체계가 창조한 처리에 결합된다. 이 처리는 접근조종을 목적으로 한 주동체이다.

만일 인증이 국부사용자등록자리 자료기지상에서 실패하면 가입요구는 다른 인증패키지를 찾아서 망우로 나갈수 있다. 만일 모든 시도가 실패하면 사용자는 오유통보문을

받게 된다. 반대로 첫 인증패키지가 국부콤퓨터가 아니라 봉사기에 있다면 두번째 시도는 의뢰기에 림시완충된 자료기지를 리용하여 진행할수 있다.

끝으로 방금 설명한 대화형가입과 사용자가 망인증규약을 통하여 가입하는 망가입사이에는 차이가 있다는것을 알아야 한다. 여러가지 접근조종조건은 사용자가 어떻게 가입하는가에 따른다.

3. SAM API의 우회

SAM자료기지는 적절한 2진형식으로 암호화된다. SMA API는 SAM에 대한 접근을 주는 조작체계의 대면부일뿐이다. 그러나 지금은 SAM파일들을 읽을수 있는 형식으로 변환하는데 리용할수 있는 프로그람들이 있다. PWDump는 그러한 프로그람들중의 하나이다. 이 프로그람은 Windows NT통과암호를 Unix체계와 공유하는 환경에서 단번서명을 제공하기 위하여 개발되였다. PWDump프로그람은 다음과 같은 정보를 돌려 준다.

사용자이름: 사용자식별자: 국부망관리자통과암호: NT통과암호: 사용자이름: 홈등록부

PWDump프로그람의 대표적인 출력은 다음과 같다.

phac105:1001:BB70C98EB15675ED78A48107248AD508:
13FD03080874168F86E5A4EF66E44C5:
Mark Curphey:\ \ fred\ profiles\ phac105\ personal:

해커는 PWDump와 같은 프로그람들을 실행시키도록 체계관리자를 기만하려고 할수 있다. WWW와 Active X 또는 MIME과 같은 전자우편쏘프트웨어들은 프로그람들을 인터네트상에서 전송하고 사용자의 워크스테이션에서 실행할수 있게 하는 특징을 가진다. PWDump를 실행가능한 내용과 함께 하나의 Web페지로 패키지화하고 체계관리자가 이 페지를 보도록 유혹함으로써 해커는 암호화된 NT통과암호를 읽을수 있는 형식으로 얻을수 있다(그러나 통과암호를 평문으로 볼수 있다고 주장하는것은 지나친 과장이다).

제4절. 접근조종 - 특징

콤퓨터체계안에 보관된 중요한 정보를 보호할 때 제1선방어는 체계자체에 대한 접근을 통제하는것이다. 제2선방어는 체계에 들어 올 허가를 가지는 체계방책성원들내에서의 접근조종이다. Windows NT에서는 이 순서로 접근조종을 취급하며 콤퓨터망을 참조한다.

1. 령역

체계에 가입하기 위해서는 그 체계에서 사용자등록자리를 가져야 한다(제7장 4절 3). 콤퓨터망의 사용자들은 어떤 다른 콤퓨터에 있는 자원이나 봉사가 요구될 때 몇번씩이나 다시 가입하는것을 바라지 않을것이다. 콤퓨터망의 관리자는 매개 콤퓨터가 제각기 보안설정을 하는것을 원하지 않을것이다. Windows NT는 단번서명과 동등한 보안관리를 보장해 주기 위해 령역을 리용한다.

령역은 공동의 사용자등록자리자료기지와 보안방책을 공유하는 콤퓨터들의 집합이다. 여기로부터 사용자들은 개별적인 콤퓨터들마다 등록자리를 필수적으로 요구하지 않고 령역으로부터 자기의 등록자리를 얻을수 있다. 령역에서 사용자등록자리자료기지의 기본복사본은 1차령역조종기(PDC)라고 하는 봉사기안에 보관된다. 사용자등록자리자료기지의 복사본들은 여벌령역조종기(BDC)에 보관된다. 사용자들은 PDC 또는 BDC에 의하여 인증될수 있다. 등록자리자료기지의 변경은 관리자가 작업하는 기계와는 무관계하게 항상 PDC주복사본에서 진행된다. BDC들에 있는 여벌복사본들은 PDC주복사본과 동시에 갱신된다. 만일 PDC를 리용할수 없다면 등록자리자료기지를 갱신하는것은 불가능하다. 따라서 PDC가 다시 리용할수 있게 되거나 BDC를 새로운 PDC로 승격시킬 때까지 기다려야 한다.

워크스테이션들은 자기의 등록자리자료기지를 가질수 있으며 동시에 령역의 성원으로도 될수 있다. 이때 사용자들은 국부자료기지로부터 자기의 허가를 취하는 국부사용자 또는 령역자료기지로부터 허가를 취하는 대역사용자로 될수 있다. 국부 및 대역등록자리를 가진 사용자는 2개의 서로 다른 보안식별자를 가진다(제7장 4절 4). 이와 류사하게 자원들도 국부적으로 또는 대역적으로 관리할수 있다. 실례로 워크스테이션에 설치된 인쇄기와 같은 자원들은 국부적으로 관리된다.

2. 가입완충기억기-공격의 잠재적인 점

어떤 워크스테이션의 사용자가 령역에 가입할 때 령역조종기를 리용할수 없다면 가입은 실패한다. 때문에 워크스테이션은 성공한 가입시도들을 완충기억시키도록 설치되고 만일 령역조종기를 찾지 못하면 이 가입완충기를 참조할수 있다. 이것은 령역조종기안의 SAM으로부터 제거된 사용자가 리용할수 있는 편리한 특징이다. 망접속을 금지함으로써 사용자는 워크스테이션이 인증을 위해 가입완충기를 리용하게 할수 있으며 여전히 체계에 접근할수 있게 된다.

이 문제를 TOCTTOU(time of check to time of use-리용시간 대 검사시간)의 특별히 시끄러운 문제로 볼수 있다. 어떤 주동체의 접근허가와 사용자권한은 사용자가 가입할 때 제정되며 전체 대화시간동안 혹은 방금 본것처럼 더 오래동안 유효하다. 때문에 허가와 사용자권한의 변경은 즉시적인 효과를 나타내지 않는다.

3. 사용자등록자리

SAM안의 사용자등록자리자료기지는 사용자에 대한 보안관련정보들을 보관한다. 사용자등록자리들은 령역사용자관리편의프로그람을 리용하여 편집한다. 사용자등록자리에서 다음의 마당들을 정의할수 있다.

- **사용자이름**: 가입을 위한 유일한 이름
- **통과암호**: 14문자길이를 가지며 암호화되여 보관된다. 사용자들이 다음번 가입에서 자기의 통과암호를 변경시킬수 있게 할수도 있고 또는 그들이 자기의 통과암호를 변경시키는것을 막을수도 있으며 통과암호에 사용기한을 설정해 놓을수도 있다.
- **가입시간과 워크스테이션** : 사용자가 언제 어느 콤퓨터에 가입하는것을 허락하겠는가를 지적할수 있다. 가입시간이 만기되였을 때 원격사용자를 봉사기로부터 강제로 분리시키기 위한 설정은 사용자를 내버리겠는가 아니면 현존하는 대화를 계속할것인가를 결정한다.
- **사용자프로필경로와 가입스크립트이름**: 프로필은 사용자의 프로그람그룹들, 망접속들, 화면색갈 등 탁상형콤퓨터환경을 정의한다. 가입스크립트는 사용자가 가입할 때 자동적으로 실행되는 묶음파일 또는 실행가능한 파일이다.
- **홈등록부**: 홈등록부가 국부콤퓨터에 있는가 아니면 망봉사기에 있는가를 지적할수 있다.
- **등록자리형**: 등록자리는 전역 혹은 국부적일수 있다. 제7장 4절 1에서 그 차이를 설명한다.
- **만기날자**: 기정값으로는 등록자리가 만기날자를 가지지 않는다.

4. 보안식별자

매 사용자, 그룹, 콤퓨터등록자리는 유일한 보안식별자번호(SID)를 가지는데 이 SID는 자유접근조종에 리용된다. SID는 등록자리가 창조될 때 만들어 지고 등록자리의 수명이 끝날 때까지 고정된다. SID의 구축에 모조(pseudo)우연입력(시계값)을 리용하므로 등록자리를 제거하고 그와 꼭 같은 파라메터로 다시 만들어도 꼭 같은 SID를 얻는것은 기대할수 없다. 이때 새로운 등록자리는 낡은 등록자리에 주어 진 접근허가를 유지하지 않는다.

령역이 만들어 지면 유일한 SID가 이 령역에 구축된다. 워크스테이션 또는 봉사기가 령역에 가입하면 령역의 SID를 포함하는 하나의 SID를 받는다(콤퓨터들이 같은 령역안에 있는가를 검사하는데 그것들의 SID를 리용한다). SID들은 변경시킬수 없기때문에 령역들사이에서 령역조종기를 옮기는것은 간단한 관리처리가 아니다. 콤퓨터는 완전히 다시 설치되고 론리적으로 <새로운> 콤퓨터로 되여 새로운 SID를 받아야 하며 새로운 령역안의 조종기로 된다.

다른 콤퓨터의 뿌리등록부와 구성파일을 복사하는 방법으로 새로운 콤퓨터를 설치하는것은 같은 SID를 가진 두대의 콤퓨터를 만들게 되므로 Windows NT보안모형에 위반된다.

5. Windows NT객체들에 대한 접근

Windows NT는 객체지향적으로 설계된 수법이다. 처리, 사용자등록자리, 자원, 파일, 등록부 등이 모두 일정한 형태의 객체들이다. 객체에 대한 자유접근조종은 객체의 형에 의존한다. 실례로 파일에로의 접근조종은 인쇄대기렬에로의 접근조종과 차이난다. 객체에 대한 접근은 주동체에 주어 진 허가를 통해서 조종된다. 매 객체는 다음의 내용들을 포함하는 보안서술자를 가진다.

- 객체소유자의 보안ID
- POSIX부분체계에 의해서만 리용되는 그룹보안식별자
- 접근조종목록(ACL)
- 발생할 검사통보를 조종하는 체계접근조종목록

ACL은 접근조종과 검사허가를 포함한다. 주동체 또는 그룹에 대한 접근조종목록기입항목(ACE)은 다음과 같다.

- AccessDenied(접근거부)
- AccessAllowed(접근허락)
- SystemAudit(체계검열)

AccessDenied(접근거부)기입항목은 ALC에서 늘 첫 항목으로 된다. 매 AccessAllowed(접근허락)기입항목은 접근허가의 목록이다. 접근허가는 객체의 형에 따라서 고유하다. 다음절에서 론의할 NTFS파일체계에 대한 접근허가는 그러한 접근허가 모임의 실례로 된다.

또한 모든 형의 객체들에 적용하는 표준접근마스크가 존재한다. 대표적실례는 다음과 같이 객체의 보안속성의 관리에 귀착되는 허가들이다.

Write_DAC: 자유ALC (AccessDenied와AccessAllowed기입항목들)을 변경
Read_Control: 보안서술자에 대한 읽기접근을 허락 또는 거부
Delete: 객체에 대한 접근허가를 주거나 거부하거나 또는 지우기

주동체가 객체에 대한 접근을 요구하면 보안참조감시기는 주동체가 요구한 접근을 허락할것인가를 결정하기 위하여 주동체의 보안접근통표(SAT)와 객체의 ACL을 취한다. 주동체의 허가로부터 요구하는 접근마스크가 구축된다.

만일 ACL이 존재하지 않으면 검사가 집행되지 않고 접근이 허락된다. 만일 ACL이 존재하면 매 ACE에 대해 주동체의 SID(SAT안에 있는)가 ACE안의 SID들과 비교된다. 다음의 3가지 경우들이 가능하다.

1. ACE는 일치하는 SID를 포함하지 않는다. 그러한 ACE는 뛰여 넘는다.

2. ACE는 《AccessDenied》를 지정하는 일치하는 SID를 포함한다. AccessDenied기입항목들이 먼저 놓이므로 그 주동체에 대한 접근이 허락된 어떤 ACE보다도 먼저 처리된다. 만일 바라는 접근마스크가 Read_Control 또는 Write_DAC 요구를 가지고 있고 주동체가 객체의 소유자라면 접근이 허락된다. 다른 경우 접근은 거부되고 더이상 검사를 진행하지 않는다.
3. ACE는 《AccessAllowed》을 지적하는 일치하는 SID를 포함한다. 만일ACE안의 접근마스크가 이미 검사된 일치하는 모든 ACE들의 접근마스크들과 함께 요구하는 접근마스크안에서 모든 허가들을 포함하면 접근은 허락되고 더이상 검사를 진행하지 않는다. 그밖의 경우에는 탐색을 계속한다.

만일 탐색이ACL의 마감에 이르기까지 접근허가를 받지 못한 상태이면 접근은 거부된다. 따라서 ACL이 비면 접근은 항상 부정되며 ACL이 존재하지 않으면 접근은 항상 허락된다.

흔히 조작체계는 접근조종정보를 여러 장소에 보관한다. 때문에 어떤 순서로 검사가 진행되는가를 아는것이 중요하다. 때때로 처음으로 일치하는 접근조종기입항목만이 참고된다. 다른 때에는 후에 나타나는 보다 더 적절한 기입항목들이 앞의 항목들을 무효화하게 된다. 끝으로 조작체계가 접근요구에 일치하는 기입항목을 찾지 못했을 때 어떻게 반응하는가를 알아야 한다.

6. NTFS파일체계

NTFS(New technology file system)에 대한 특정한 접근허가는 구동기, 등록부, 파일들에 대한 접근을 제한한다. 파일의 소유자가 다른 사용자나 그룹에 대하여 접근허가들을 정의한다. 기본적인 허가들은 다음과 같다.

- 읽기(X)
- 쓰기(W)
- 실행(X)
- 지우기(D)
- 접근허가의 변경(P)
- 소유권가지기(O)

다음의 접근허가들이 파일들에 적용된다.

- NoAccess: 어떤 접근도 막으며 사용자가 그룹성원자격을 통해서 가질수 있는 그 어떤 다른 접근도 무효로 한다.
- Read(RX): 읽기 및 실행접근만 허락한다.
- Change(RWXD): 읽기, 쓰기, 실행 그리고 지우기접근을 허락한다.
- Full Control(all): 파일의 읽기,쓰기, 실행, 지우기를 허락하며 접근허가를 변경시키는것과 소유권을 가지는것을 허락한다.

- Special Access: **파일의** 읽기, 쓰기, 실행, 삭제허가와 접근허가의 변경, 소유권획득의 어떤 조합이 될수 있다.

사용자의 접근허가들은 사용자마다 개별적으로 설정한 허가들로부터 그리고 그 사용자가 속해 있는 모든 그룹들의 허가들로부터 유도된다. 다만 《NoAccess》사용자허가만은 관련된 그룹허가들을 무효로 한다. 한편 사용자는 자기가 직접 지적한것보다 더 많은 허가를 가질수 있다.

리상적으로는 조직적보안방책이 동일한 요구를 가지는 사용자들을 관리하기 쉬운 일련의 그룹으로 가른다. 실천에서는 항상 례외가 있게 된다. 따라서 례외를 정의하는데서 허가들을 회수하거나 새로 첨부하는 기구들은 적절하게 사용하면 쓸모 있는 도구로 된다.

새로운 파일이 창조되고 NTFS허가들이 어미등록부에 적용되였다면 그 파일은 어미등록부로부터 접근허가들을 상속 받게 된다.

한편《Everyone》은 새로 창조된 모든 파일들에 대해 《Full Control》허가를 가진다. 등록부들에 대한 허가는 다음과 같다.

- NoAccess: 등록부에 대한 어떤 접근도 막으며 사용자가 그룹내에서의 성원자격을 통해서 가질수 있는 어떤 다른 접근도 무효로 한다.
- List(RX): 사용자는 등록부와 그의 부분등록부안에 있는 파일들을 렬거할수 있다.
- Read(RX): 사용자는 등록부와 그의 부분등록부안에 있는 파일들을 렬거하거나 열수 있고 부분등록부안으로 옮기거나 응용파일들을 실행할수 있다.
- Add(WX): 사용자는 파일들을 등록부에 추가할수 있으나 등록부의 내용을 볼수는 없다.
- Add and Read(RWX): 사용자는 파일들을 보거나 열수 있고 응용파일들을 실행할수 있으며 파일들과 부분등록부들을 추가할수 있다.
- Change(RWXD): 사용자는 파일들을 읽기, 쓰기, 실행, 삭제할수 있으며 파일들과 부분등록부들을 추가할수 있다.
- Full control(all): ACL들에 대한 고려없이 부분등록부들을 비우거나 파일들을 지우기할수 있는 허가를 포함하여 등록부우에서의 완전한 접근을 가진다.
- Special Directory Access: 등록부허가들의 어떤 조합이 될수 있다.
- Special File Access: 파일접근허가들의 어떤 조합의 창조를 허용한다.

파일이 등록부들사이에서 이동할 때 그의 NTFS허가들은 보존된다. 파일이 복사될때에는 목적등록부의 허가를 넘겨 받는다. 등록부에 대한 지우기허가를 가진 사용자는 그 등록부내에 있는 어떤 파일도 지우기할수 있으며 파일 그자체에 대한 지우기허가를 따로 요구하지 않는다.

검열해야 할 사건들은 읽기, 쓰기, 실행, 삭제, 접근허가의 변경, 소유권획득이다. 관리자는 어느 사건을 감시하겠는가를 결정할수 있다.

7. 공유

공유란 망통신규약들과 조작체계들이 분리된 실체였던 시대의 유물이다. 공유는 망사용자(대화형사용자와 혼돈하지 말것)가 파일이나 등록부에 어떻게 접근할수 있는가를 조종한다. 사용자는 읽기만을 위해서 또는 완전접근을 위해서 통과암호를 설정함으로써 《공유된》 파일에 대한 망접근을 허락할수 있다. 이 도식은 통과암호에 기초한 접근조종의 모든 결함을 그대로 가지게 된다. 즉

- 서로 다른 통과암호들이 서로 다른 공유자원들에 기억되여야 한다.
- 공유자원에 대한 통과암호는 접근을 허가할 모든 사용자들에게 공유되여야 한다.

더우기 공유들은 실제로 접근요구들의 검열을 지원하지 않는다. 망사용자들은 공유들을 통해서뿐아니라 그들이 접근하려고 하는 체계에 설정된 NTFS허가들을 통해서도 조종된다.

제5절. 접근조종-관리

앞에서는Windows NT의 본질적인 접근조종기구에 대하여 보았다. 여기서는 보안관리의 문제에로 방향을 돌리기로 한다. 내장된 등록자리, 위임프로필, 령역, 신용관계등은 보안관리를 보다 쉽게 할수 있는 개념들이다.

1. 국부 및 전역그룹

그룹이란 사용자등록자리들의 집합을 말한다. 그룹의 성원들은 그 그룹에 주어 진 사용자등록자리들과 허가들을 계승한다. 그룹들을 조종의 중간층으로 볼수 있다. 객체들에 대한 허가는 그룹에 주어 진다. 사용자는 이 그룹의 성원으로 됨으로써 객체에 대한 접근허가를 가지게 된다. 그룹에 주어 진 허가들은 이 그룹의 개별적성원들로부터 선택적으로 회수될수 있다. 만일 어떤 성원이 어떤 특별한 파일에 접근하지 말아야 한다면 체계관리자는 그 파일에서 그 사용자를 위한 허가를 NoAccess로 설정하여야 한다.

령역내에서 그룹들은 령역전체에 대해서 포괄적으로 정의되거나 혹은 개별적워크스테이션에 대해서 국부적으로 정의될수 있다.

- **전역그룹**: 령역에 대해서 정의되며 사용자등록자리만을 포함하고 그밖의 그룹등록자리는 포함하지 않는다.
- **국부그룹**: 워크스테이션에 대해서 정의되며 사용자등록자리와 전역그룹을 다 포함한다.

전역그룹들과 국부그룹들은 주동체와 객체들사이에 두개의 조종층을 설치하는 수단을 제공한다(그림 7-2). 전역그룹들은 동일한 접근권한을 가지는 사용자들을 묶는다. 어떤 기계에 있는 객체들에 대한 허가는 개별적사용자등록자리들이 아니라 그 자원들에 접근해야 하는 전역그룹들을 포함하는 국부자원그룹에 주어 진다.

이러한 그룹구조에 대한 해제는 두가지 방법으로 정의될수 있다. 앞서 본것처럼 《NoAccess》 허가를 통하여 그룹성원으로부터 허가들을 회수할수 있다. 사용자등록자리를 관계되는 국부그룹안에 배치함으로써 사용자에게 보충적인 허가들을 줄수 있다.
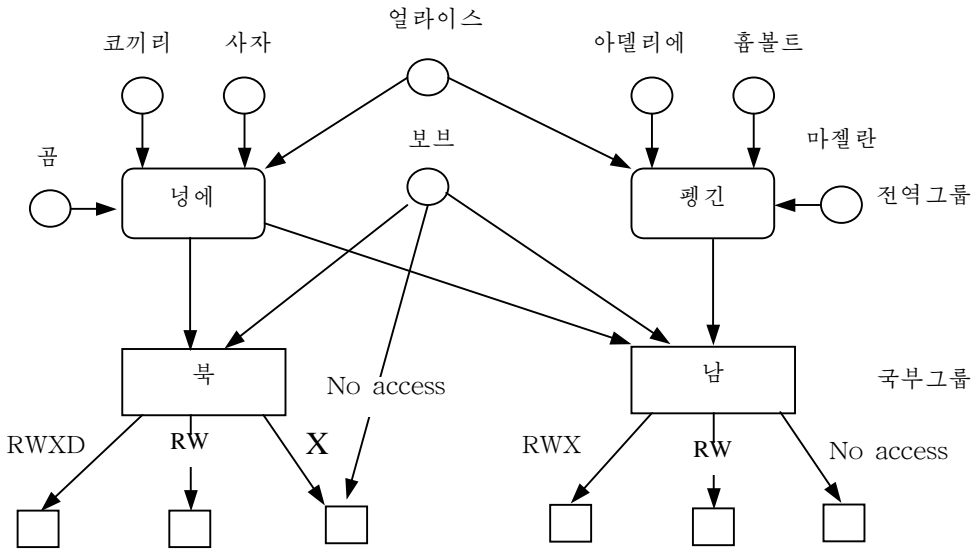
그림 7-2. Windows NT에서 전역 및 국부그룹

2. 사용자권한

사용자권한은 개별적인 객체들에 대한 접근을 조종하지 않는다. 오히려 사용자가 체계에서 무엇을 하도록 허락되였는가를 지적한다(다른 조작체계들에서는 류사한 개념들을 특권이라고 한다).

대표적인 사용자권한은 검사, 여벌복사, 체계끄기와 같은 체계관리작용을 할 권리이다. 국부적 또는 망상에 있는 콤퓨터에 대한 접근도 사용자의 권한으로 조종할수 있다. 더우기 소유자로 하여금 접근조종을 우회하게 하는 사용자권한도 있다. 객체의 소유권을 획득하기 위한 권한을 실례로 들수 있다.

표준접근조종절차에 따르면 어떤 등록부에 대한 접근이 부정된 사용자는 그 등록부안의 어떤 파일의 ACL에 의하여 접근이 허락되였다 해도 그 파일에 접근할수 없다. 그러나 등록부경로에서 ACL들은 검사를 우회할 권한을 가진 사용자에 대해서는 검사를 하지 않는다. 기정값으로 모든 사용자가 이 권한을 가진다. 보다 명백한 접근조종방책을 가지기 위해 사용자와 그룹들로부터 이 권한을 제거할것을 권고한다.

3. 기성그룹

기성등록자리들과 그룹들은 미리 정의된 사용자의 권한과 허가들을 가진다. Domain Admins(령역관리자), Domain Users(령역사용자), Domain Guest(령역손님)와 같은 몇개의 전역적인 기성그룹들이 존재한다.

대부분의 기성그룹들은 대체로 Administrators, Backup Operators, Users, Guest와 같은 국부그룹들이다.
체계관리자들은 보안방책을 실현할 때 될수록 기성그룹들을 리용하는것이 좋다. 그리고 꼭 그렇게 해야 할 리유가 있을 때에만 여러가지 허가패턴을 가진 그룹들을 정의해야 한다.

보안전문가가 아닌 사용자들이 보안특징차림표에서 전문가적인 선택을 할수는 없다. 이러한 상황에서는 표준요구에 따라 미리 정의된 보안특징설정을 리용하는것이 좋다.

Windows NT에서 체계관리는 기성국부그룹의 관리자로 할당된 사용자가 수행한다. 기성등록자리 Administators는 Windows NT가 설치될 때 창조된다. 이 등록자리는 자동적으로 국부Administrators그룹의 한 성원으로 되며 제거할수도 없고 금지시킬수도 없다.
Unix에서와는 달리 Administrators는 모든 파일에 접근할수 있는 특권사용자특권을 자동적으로는 가지지 못한다. Administrators가 자동적인 접근을 할수 없게 파일에 대한 허가를 설정할수 있다. 정상적인 체계관리작용은 Server Operators, Backup Operators, Account Operators, Print Operators와 같은 기성국부그룹들에 할당된 등록자리들로부터 수행되여야 한다. 이 그룹들의 권한은 그것들의 과제에 맞게 제한된다. 워크스테이션에서 Power User등록자리는 인쇄기나 국부등록자리와 같은 국부자원들을 취급할수 있다.
이와 같이 접근권한들을 세밀하게 분할했어도 Administrator등록자리는 여전히 자기에게 적용된 제한들을 우회할수 있는 지위에 있다. Backup과 같은 적은 권한을 가진 기성국부그룹들조차 Backup이 현존하는 파일보호를 무시하므로 접근조종을 우회할 기회를 가진다.
만일 사용자가 체계관리자로서 작업한다면 이 사용자의 등록자리를 Domains Admins기성국부그룹안에 넣고 이 국부그룹을 Administrators안에 넣는다. 이것은 사용자등록자리를 직접 국부관리자그룹에 배치하는데 리용할수 있다. 체계관리자로서 작업하는 사용자는 보안의 리유로 하여 Users국부그룹에 배치할수 있는 예비등록자리를 가지고 있어야 한다. 그러나 사용자가 한 등록자리에서 다른 등록자리에로 변경할 때 탈퇴하고 다시 가입해야 하므로 불편한 점이 있다.
Guests등록자리는 통과암호를 요구하지 않으며 사용자에게 인증을 요구하지 않는 자원에 대한 접근을 줄 때 리용할수 있다. 그러나 허가는 다른 사용자등록자리에서처럼 이 등록자리에 주어 질수 있다. Windows NT가 설치될 때 Guests등록자리는 금지된다.
이외에도 접근허가들을 효과적으로 정의하는데 리용할수 있는 기성그룹들이 더 존재한다.

- **Everyone**: Guests를 포함하여 모든 국부 및 원격사용자들을 포함한다. 이 그룹은 모든 사용자들에 대한 허가를 허락 및 부정하는데 쓴다.
- **Interactive**: 국부적으로 가입한 모든 사용자들을 포함한다.
- **Network**: 망상에서 가입한 모든 사용자들을 포함한다.
- **System**: 조작체계

- Creator Owner: 파일 또는 자원의 창조자나 소유자

4. Windows NT에서 신용관계

중간조종층들은 보관관리를 효과적으로 하게 한다. 조직이 클수록 중간층은 더욱 편리하며 전역그룹들이나 국부그룹들 그리고 기성등록자리들만으로는 보관관리를 충분히 할수 없다. 여기서는 보안조종의 다음단계로서 령역을 리용하고 령역들사이에 신용관계를 설정한다.

한방향신용관계에서는 신용 받는 령역(trusted domain)과 신용하는 령역(trusting domain)을 가진다. 신용 받는 령역으로부터의 사용자등록자리는 신용하는 령역에서도 유효하다. 두방향신용관계는 2개의 한방향신용관계를 설정하여 수립할수 있다. 신용관계들은 이동성을 가지지 않는다. 즉 령역 A가 령역 B를 믿고 령역 B가 령역 C를 믿는다고 해서 령역 A가 령역 C를 믿는것은 아니다. 기술적으로 신용관계를 다음과 같이 설정할수 있다.

- 신용 받는 령역의 관리자는 신용하는 령역의 이름을 지적하는 중간령역신용등록자리를 설정하고 이 령역의 통과암호를 선택한다.
- 이 통과암호는 신용하는 령역의 관리자에게 주어 져야 한다.
- 신용하는 령역의 국부보안권(LSA)은 신용 받는 령역의 이름과 SID를 포함하는 신용 받는 령역객체를 창조하고 신용 받는 령역의 관리자에게서 수신한 통과암호를 포함하는 비밀객체를 만든다.
- 그다음에 신용하는 령역의 LSA는 중간령역신용등록자리를 리용하여 신용 받는 령역에로 가입하려고 시도한다. 이 시도는 중간령역등록자리를 가입에 리용할수 없으므로 실패를 전제로 한것이지만 귀환된 오유통보문은 등록자리가 존재한다는것을 확증해 준다. LSA는 계속하여 후에 신용 받는 령역으로부터의 접근요구를 처리할 때 요구될 신용 받는 령역의 령역조종기에 대한 정보를 갱신한다.
- 신용받는 령역과 신용하는 령역사이에 공유된 통과암호는 신용하는 령역안에 있는 PDC에 의해서 관리되며 규칙적으로 갱신된다. 통과암호는 사용자인증이 진행되는 동안 부정거래를 막기 위해 리용된다.

신용관계들은 어떻게 리용될수 있는가? 실례로 어떤 조직에서 모든 사용자등록자리들을 하나의 단일등록자리령역에 넣을수 있다. 서로 다른 구역들에 소유된 자원들은 자원령역들에 배치된다. 다음에 신용 받는 령역으로는 등록자리령역을 그리고 신용하는 령역들로는 자원령역을 가지고 한방향신용관계를 수립한다.

그림 7-3의 실례에서 사용자 Alf에게 자원 Colour Printer에 대한 통제된 접근을 주기 위하여 다음과 같이 한다. 등록자리령역에 있는 전역그룹 Researchers에 Alf를 넣는다. 자원령역PRINTERS안에 국부그룹 Colour Printer를 정의한다. 신용 받는 령역으로서 EMPLOYEES와 신용하는 령역으로서 PRINTERS를 가지고 신용관계를 수립한다. 신용하는 령역에 있는 국부그룹 Colour Printer에 신용 받는 령역의 전역그룹 EMPLOYEES\ Researchers를 추가한다. 사용자 Alf가 접근을 요구할 때 신용하는 령역에 있는 LSA는 신용 받는 령역에 있는 령역조종기로부터 사용자의 권한들과 허가들을 되찾게 된다.
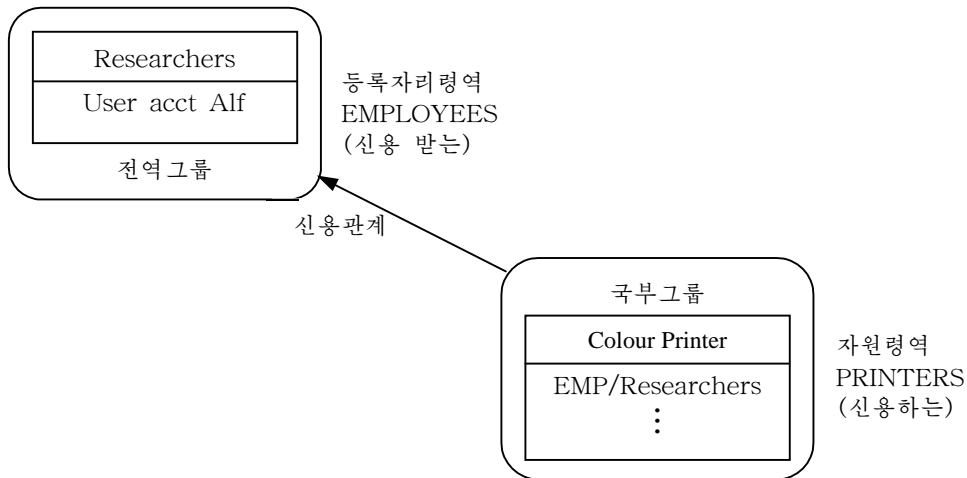
그림 7-3. 령역들사이의 신용관계

어떤 조직내에서 중심화된 사용자등록자리와 체계관리는 사용자등록자리자료기지와 관리자그룹들을 포함하는 주령역을 창조함으로써 실현될수 있다. 이 주령역은 다른 모든 령역들에 의해 신용 받으며 따라서 주령역에서 등록자리를 가지는 사용자들은 다른 모든 령역들에 접근할수 있다. 주령역으로부터 관리자그룹들은 국부령역에 있는 각각의 그룹들에 배치된다. 령역에 대한 접근을 얻기 위해서 그 령역에서의 등록자리를 요구하거나 또는 그에 의해 신용 받는 령역에서의 등록자리를 요구하게 된다. 이 규칙에서 한가지 례외가 있다. 다른 령역의 워크스테이션들에 대한 접근을 얻자면 그 워크스테이션에 관한 국부등록자리를 자신의 정규령역등록자리에서와 꼭 같은 이름으로(통과암호도) 창조할수 있다. 같은 이름을 가지는데도 불구하고 이 두개는 서로 다른 등록자리이며 워크스테이션은 일치하는 등록자리를 가지기만 하면 접수할것이다. 두개의 등록자리들이 서로 다른 통과암호를 사용한다면 국부등록자리에 관한 통과암호를 입력해야 한다.

5. 위임된 프로필

사용자의 프로필은 사용자의 탁상콤퓨터환경 즉 특별히 사용자가 호출할수 있는 프로그람을 정의한다. 위임된 프로필(mandatory profile)은 사용자가 변화시킬수 없다. 위임된 프로필은 사용자의 탁상콤퓨터환경에 제공된 편의프로그람들 즉 Program Manager 을 제한할수 있기때문에 하나의 보안기구이다. 관리자는 사용자의 탁상콤퓨터환경에서 리용할수 있는 특징들을 정의하는 사용자프로필안에 제한들을 설정할수 있다.

제한들은 EditLevel, Noclose, NoFileMenu, NoRun, Nosave, Settings Restrications, Show Command Groups 에 지정할수 있다. EditLevel을 실례로 들면 사용자가 자기의 Program Manager를 변경하는 방법을 제한한다. EditLevel은 다음의 값들을 가질수 있다.

0: 모든 변경을 허락한다. 이것은 기정설정값이다.
1: 사용자가 그룹의 창조, 지우기, 다시 이름짓기를 막는다.
2: EditLevel=1의 제한을 포함하며 사용자가 프로그람항목을 만들거나 지우는것을 막는다.

3: EditLevel=2의 제한을 포함하며 사용자가 프로그람항목의 지령행을 변경시킬수 없게 한다.
4: EditLevel=3의 제한을 포함하며 사용자가 어떤 프로그람항목의 자료를 변경시킬수 없게 한다.

제6절. 검 열

Windows NT는 검열기록을 보존한다. 검열기록안의 항목들은 보안참조감시기(Security Reference Monitor)에 의해서 발생된다. 검열기록에 기록된 보안관련사건들은 정당한 또는 부정당한 가입시도들, 특권리용, 자원(또는 파일)의 창조, 지우기, 열기와 같은 사건들을 포함한다. 기록할 사건들은 User Manager의 Policies Menu와 FileManager의 Security Menu에서 선택할수 있다. 관리자만이 이 봉사프로그람들을 실행시킬수 있으며 검열기록을 관리할수 있다. 검열기록은 Event Viewer에 의하여 조사된다.

검열기록의 최대크기는 Event Viewer에서 설정할수 있다. 기록이 그의 최대크기에 이르면 Windows NT는 다음의 3가지 선택적인 대책을 요구한다.

- **Overwrite events as needed**: 검열기록안에서 제일 오랜 항목우에 덧쓰기를 한다.
- **Overwrite events older than[] days**: 지적된 한계보다 더 오랜 기입항목들우에 덧쓰기한다.
- **Do not overwrite events**: 기록을 새 사건을 기록하기에 앞서 수동적으로 지워야 한다.

체계관리자는 마지막 두개를 선택하여 체계를 설정함으로써 기록이 가득 찼을 때 자동적으로 콤퓨터가 꺼지도록 할수 있다. 등록고안에 있는 CrashOnAuditFail 기입항목은 다음과 같이 설정된다.

HKEY_LOCAL_MACHINE\ SYSTEM\ CurrentControlSet\ Control\ Lsa
Name: CrashOnAuditFail
Type:REG_DWORD
Value:1

이 설정은 오렌지부크 C2요구들에 맞추기 위해 필요할수 있다.

제7절. DLL의 보안측면

동적련결서고는 프로그람이 실행될 때 련결되는 쏘프트웨어모듈들이다. DLL코드는 그의 주프로그람의 권리를 가지고 실행된다. 트로이목마코드는 트로이목마 DLL에로 프로그람을 유인하든가 아니면 원래의 DLL자체를 수정하여 끼워 넣을수 있다.

1. DLL기만

련결알고리듬은 DLL의 위치를 지정하기 위한 세가지 선택을 가지고 있다. 알고리듬은 다음과 같은것들을 탐색할수 있다.

- 프로그람등록부: 주프로그람의 실행가능한 파일을 보관하고 있다.
- 체계등록부
- 작업등록부: 주프로그람을 호출하는 처리의 현행등록부

첫 두개의 장소들은 미리 알려 져 있어서 DLL에로의 접근을 보호하기 위한 적당한 대책을 취할수 있다. 문제는 마지막선택에서 발생한다. 만일 련결알고리듬이 작업등록부를 탐색한다면 공격자는 원래의 DLL과 꼭 같은 이름을 가진 트로이DLL을 사용자가 작업등록부로 쓸수 있는 등록부에 배치할수 있을것이다. 이 공격을 막기 위한 대책을 위해서는 Windows NT를 보다 구체적으로 고찰해야 한다.

2. 통보부분품

통보부분품(notification Package)은 다른 조작체계와 망제품들이 Windows NT와 함께 실행하는 환경에서 단번서명을 돕기 위해 도입된 기구이다. 통과암호를 공유하는것을 허락하기 위해 통과암호를 해득해야 하며 그다음에 평문통과암호가 통보부분품이라고 부르는 DLL에 의하여 취급되여야 한다. 사용중에 있는 DLL은 잠그어 져 있으며 보호되여 있다. 사용하지 않는 DLL과 모든 사용자가 변경허가를 가지는 등록부안에서 있는 DLL은 평문통과암호를 획득한 트로이목마코드로 바뀔수 있다.

이 장의 문헌안내

특정한 조작체계에 관한 대부분의 보안편람들은 보안체계의 겉면만을 취급하고 있다. 이것들은 제공된 특징들과 관리측면에 초점을 두고 있다. 이 편람들은 독자들에게 체계보안을 어떻게 관리하는가가 아니라 보안체계특징들을 관리하는 방법을 가르쳐 주고 있다.

초보적인 준위에서 Windows NT보안을 취급하며 많은 련습을 포함하는 참고문헌은 [149]이며 [63,139]들은 이미 조작체계를 많이 다루어 본 독자들에게 적합하다. 마이크로쏘프트의 지식기지와 마이크로쏘프트의 다음의 Web페지는 조작체계의 보안특징들, 보안경보, 관할구역들에 대한 정보를 포함한다.

http://www.마이크로쏘프트.com/security

약점을 분석하는 Windows NT보안의 NSA평가와 적절한 대책에 관한 지도서는 다음주소로부터 찾아 볼수 있다.

http://www.TrustedSystems.com

Windows NT보안에 관한 기타 유용한 자원들은 다음주소들에 있다.

http://www.ntshop.net
http://www.ntresearch.com/link.htm

Windows NT이외에 재정분야에서 대중적인 안전조작체계는 참고문헌[163]에서 제시한다. RACF에 대한 정보, IBM의 Resource Access Control-Facility(자원접근조종기구)에 관한 정보는 다음페지에서 찾을수 있다.

http://www.s390.ibm.com/racf

AS/400(Application System 400)은 중규모콤퓨터들을 위한 IBM조작체계이다. 이 조작체계는 객체지향적인 조작체계로서 통속적이며 상당히 안전한 체계로 리용되였다. AS/400 의 보안에 대해서는 참고문헌[122]에서 참고할수 있다.

련습문제

1. 등록고에 대한 호출은 등록고편집기에서 접근허가를 설정하거나 등록고열쇠에 대한 접근허가를 정의함으로써 조종할수 있다. 이 두 방법에 의한 보호결과를 비교하시오.
2. Windows NT에서는 BIOS통과암호를 정의할수 있다. 이 통과암호는 오직 수동적으로만 입력할수 있다. 이 사실로부터 통과암호선택에 관하여 어떤 요구가 나서는가?
3. Windows NT에서 령역은 워크스테이션과 령역조종기로 구성된다. 관리자등록자리와 령역안의 모든 콤퓨터들은 같은 통과암호를 가져야 하는가?
4. Unix통과암호와 Windows NT LAN Manager 통과암호들을 블로크암호로 암호화하는 방법들을 비교하시오. 암호화된 LAN Manager통과암호는 해득할수 있지만 Unix통과암호는 해득할수 없다. 겉으로 보기에는 비슷한 알고리듬이 왜 이렇게 차이나는 속성을 가지는가?
5. Unix와 RACF 그리고 Windows NT에서 접근권한은 사용자와 그룹에 대하여 정의된다. 더 좋은 보안관리를 위하여 사용자들은 그룹으로 묶어 질수 있다. 이 3개의 조작체계는 사용자가 그룹보다 낮은 특권을 가질 때 접근요구를 어떻게 결정하는가? 그룹에 주어 진 접근권한이 어떻게 개별적인 성원들에게서는 억제 당하는가?
6. 프로그람을 실행시키려면 실행가능한 프로그람파일에 어떤 허가가 요구되는가? 사용자가 자기의 복사본을 만드는것을 어떻게 막을수 있는가? 사용자들이 자기의 복사본을 실행시키게 하면 어떤 보안위험이 생길수 있는가?
7. 신용관계는 이동적인가? Windows NT의 령역들사이 신용관계와 Unix체계의 .rhosts 를 비교하시오.
8. Windows NT4.0 에서 령역은 약 10000개의 사용자등록자리를 지원할수 있다. 만일 체계가 더 많은 사용자들과 주령역모형의 원리들을 취급해야 한다면 어느 령역과 신용관계를 설정해야 하는가?
9. 사용자에게 원격워크스테이션에 대한 접근을 주기 위하여 정합등록자리를 리용할 때 무엇이 관리적이며 보안과 관련되는가?
10. DLL침입을 막는데 리용할수 있는 방어방법을 연구하시오.
11. Windows NT는 많은 측면에서 Unix보다 많은 보안특징들을 제공한다. 일부 비평가들은 사용자들이 너무 많은 보안특징들로 하여 복잡해 질수 있으며 결과적으로 쓸모가 없다고 주장한다. 보안특징들을 무엇을 지향하여 확장하여야 하는가? 만일 특징이 너무 많거나 너무 적다면 어떤 문제가 생기겠는가?

제8장. 보안실패의 원인과 대책

콤퓨터보안은 무엇인가 잘못되면 통보를 내보내거나 오동작하게 된다. 보안오유의 원인을 마지막까지 추적하면 대체로 다음과 같은 3가지 원인중의 하나라는것을 알수 있다.

- 변경(change)
- 자만(complacency)
- 편리(convenience)

이 장은 흔히 있는 실수(mistake)들을 반복하지 않도록 경고를 주는것을 목적으로 한다. 콤퓨터비루스에 대한 부분 즉 콤퓨터보안에서 가장 눈에 뜨이는 콤퓨터위험의 측면들을 취급한다.

목적

- 대부분의 보안오유를 일으키는 기본원인들을 제시한다.
- 실수들이 자주 되풀이되고 있다는것을 인정한다.
- 보안체계를 설계할 때 취할 예방책에 대하여 리해한다.
- 콤퓨터비루스와 항비루스쏘프트웨어에 대하여 소개한다.

제1절. 소 개

여기서는 콤퓨터보안이 접근조종방책를 시행한다는것을 전제로 한다. 이것은 다음과 같은 두가지 측면을 의미한다.

1. 우선 응용분야의 보호요구를 반영하는 접근조종방책을 형식화하여야 한다.
2. 콤퓨터체계는 조종을 회피하거나 무효화하려는 적극적인 시도가 있을 때 그 방책을 시행하여야 한다.

복잡한 체계를 실현하는것은 어려운 과제이며 조작체계에는 보안결함들이 있다는것을 예상하여야 한다. 그런데 놀랍게도 이러한 결함들의 원인을 보면 흔히 매우 단순한 프로그람작성실수들이며 많은 공격들은 교묘한 기교나 깊은 기술적지식에 의해서가 아니라 잘 알려 진 보안약점(또는 설계특징)들을 리용한다는것이다. 누군가에게서 얻은 도구를 가지고 체계의 약한 점들을 탐색하는 공격자들은 보안관리자의 사업을 어렵고 시끄럽게 한다. 보안문제들의 주요원인은 다음부류들에 해당한다.

- 환경변화
- 경계와 문장론검사
- 편리하지만 위험한 설계특징
- 통제된 호출(Controlled Invocation)로부터의 탈퇴
- 낮은 층에서의 우회
- 통신규약실현에서의 빈틈

몇가지 실례들을 통해 이 문제들을 보기로 하자. 실례들은 널리 알려 진 정보들을 사용하도록 선택하였다. 일부 조작체계들을 자주 언급한다고 해서 그것들이 완벽한 보안을 실현한것으로 오해하지 말아야 한다.

제2절. 환경의 변화

보안에서 가장 위험한 적수들중 하나가 변경이다. 다음의 두가지 실례들은 변경의 영향을 보여 준다.

1. 미치광이해커

첫 사건은 1987년부터 시작되여 1991년 영국에서 콤퓨터해커에 대한 첫 유죄판결을 내리는것으로 끝났다. 해커가 침입한 조작체계는 ICL의 VME/B였다.

VME/B는 파일서술자에 파일들에 대한 정보를 기억한다. 모든 파일서술자들은 사용자 :STD에게 소유된다. 이것은 일단 파일서술자들이 서로 다른 보안준위들에서 비밀에 관계될수 있다면 문제를 발생시키게 한다. 그 리유로 :STD는 비밀에 관계되는 파일서술자들에 접근하지 않았다. 따라서 이 서술자들은 정상적인 여벌만들기기간에 회복될수 없다. 해결책은 명백하였다. 비밀에 관계되는 파일서술자들을 소유한 새로운 사용자 :STD/CLASS가 만들어 졌다. 다음 이것은 체계가 갱신하는 루틴에 포함된다.사용자 :STD/CLASS는 파일서술자들을 소유하는외에 다른 목적을 가지지 않았다. 따라서 그 누구도 :STD/CLASS로 가입할수 없으며 또 그럴 필요도 없었다. VME/B설계자들은 :STD/CLASS의 통과암호를 RETURN건으로 정의함으로써 가입을 불가능하게 하려고 하였다. RETURN은 항상 통과암호의 경계기호로 해석되며 통과암호의 부분으로는 되지 않기때문에 누구도 가입할수 없다. 사용자프로필에 통과암호를 설정하는것은 16진코드를 덧붙이는것으로 수행되였다. 그런데 공교롭게도 다른 마당이 변경되여 가입할수 없는 사용자대신에 인식할수 없는 보안준위를 가진 사용자가 출현하였다. 이 인식할수 없는 보안준위는 《무제한》으로 해석되므로 설계자들은 자기들의 목적과 반대결과를 얻었다.

아직도 한개의 방어선이 남아 있다. 사용자 :STD/CLASS는 오직 주조종탁에서만 가입할수 있다. 그러나 일단 주조종탁이 꺼지면 련결을 여는 다음장치가 주조종탁으로 취급되게 된다.

이 결함이 VME/B체계 그자체를 관리하고 있던 해커에게 악용되였다. 이리하여 해커는 체계에 대한 상세한 해석과 경험을 위한 충분한 기회를 가지였다. 해커는 콤퓨터쎈터가 일을 보지 않는 밤시간에 전화회선을 통하여 대학의 콤퓨터들에 침입하여 체계와 사용자파일들을 수정하거나 지우고 《Mad Hacker》라는 통보문을 남겨 두었다. 후에 그는 정확히 추적되여 법정에 끌려 가 유죄파결을 받고 감금형을 언도 받았다.

2. CTSS

다음이야기를 하기전에 초기의 시분할조작체계의 하나였던 CTSS에서 있었던 보안결함에 대하여 보자.

한번은 어떤 사용자가 통과암호파일이 《message of the day》로 주어 졌다는것을 발견하였다.

어떤 일이 일어 났는가? CTSS에서는 모든 사용자가 동일한 홈등록부를 가지였다. 한 사용자가 편집기를 호출하면 새 파일이 이 등록부에 창조된다. 이 새 파일은 SCRATCH라는 고정된 이름을 가지는데 이것은 편집되는 파일이름과는 독립이다. 이것은 한 사용자가 한순간에 오직 하나의 응용프로그람밖에 실행할수 없으므로 사리에 맞는 설계방안이였다. 그외에는 누구도 다른 사용자의 등록부에서 작업할수 없었다. 그러므로 그 편집기를 위해서 하나이상의 새 파일을 제공할 필요는 없었다. 여기까지는 그런대로 좋았다. 더 나아가서 체계가 자기의 고유한 등록부를 가지는 사용자로 취급되였다. 일부 단계에서 몇몇 사용자들은 체계관리자들로서 작업하고 있었다. 동시에 하나이상의 체계관리자가 작업하는것을 허락하는것(체계등록부에 접근)이 편리한것 같았다. 이 특징은 다음과 같이 실현되였다.

1. 한 체계관리자가 날자통보문을 편집하기 시작한다.
   SCARTCH:=MESS
2. 두번째 체계관리자가 통과암호파일을 편집하기 시작한다.
   SCARTCH:=PWD
3. 첫번째 관리자가 편집된 파일을 기억한다.결국
   MESS:=SCRATCH=PWD를 얻는것으로 된다.

이 사건렬을 그림 8-1에 보여 준다.
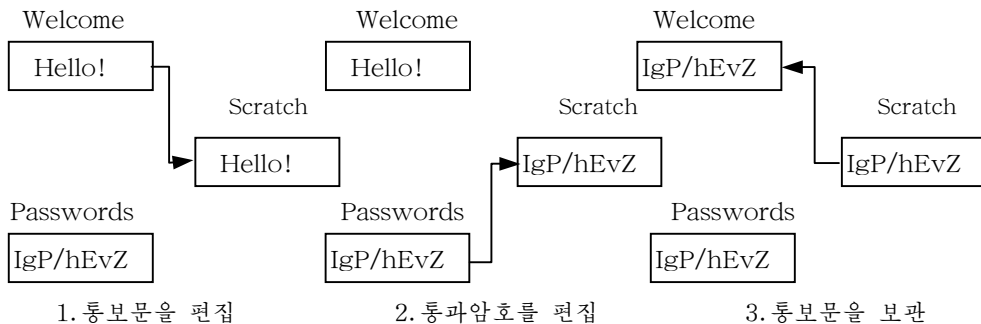
그림 8-1. CTSS에서 새 파일의 공유

제3절. 경계와 문장론검사

보안문제들에서 많이 나타나는 오유원천은 인수들의 크기나 문장론을 검사하지 않는 지령들이다. 상세한 체계지식이 있는 공격자는 입력완충기를 넘쳐 나게 함으로써 보안관련자료를 가지고 있는 기억기위치를 덧쓰기할수 있다.

I. Finger지령바그

UNIX의 **Finger**지령은 그리 좋은 보안평판을 받지 못하였다. 여기서 공격자에게 가치가 있는 정보에 접근하는 능력에 대해서는 살펴 보지 않고 1988년의 《Internet Worm》에 의하여 Unix 4BSD를 실행하는 VAX체계들에 침입하는데 악용된 바그에 대하여 본다.

Fingered데몬(daemon)은 원격 **Finger**요구들에 봉사하는 배경프로그람이다. 이 데몬은 입력의 길이를 검사하지 않으며 입력완충기를 위하여 **Gets**서고루틴을 리용한다. 완충기넘침은 다른 기억위치들을 덧쓰기한다. 만일 조작체계가 기억기를 어떻게 할당하는지 알고 있다면 넘침(overflow)에 의해 감염되는 기억기위치들을 조종할수 있다. 이와 같은 공격은 특수한 536-bit통보문으로 완충기를 넘쳐 나게 하며 체계탄창을 덧쓰기하여 기본루틴의 탄창을 변경시킨다. 이때 탄창에 씌여 진 명령들을 상세하게 보면 다음과 같다[146].

```
Pushl $68732f '/sh\ 0'
Pushl $6e69622f '/bin'
Mov sp,r10
Pushl $0
Pushl $0
Pushl r10
Pushl $3
Movl sp,ap
Chmk $3b
```

탄창은 기본루틴에로의 귀환에서 TCP 를 통하여 원격쉘에로의 접속을 여는 지령

execve("/blulsh",0,0)

이 실행되도록 설정되였다. 이 공격은 서고루틴 **gets** 가 완충기의 자리넘침을 막기 위해 입력렬의 길이를 검사한다면 불가능하게 된다. 겸하여 말하면 **gets**가 이러한 문제를 Unix루틴에서만 일으키는것은 아니다.

2. VMS가입

디지털회사의 VMS 조작체계의 한 판본에는 가입절차에 바그가 있었다. 사용자이름을 재촉할 때 사용자는 다음지령을 입력하여 접근하려는 기계를 지정할수 있다.

username/**DEVICE** = <machine>

이때 인수 machine의 길이를 검사하지 않는다. 만일 장치이름이 너무 길면 가입에 의해서 기동된 처리의 특권마스크가 장치이름으로 덧써지게 되며 사용자에게 자기의 특권준위를 설정할 능력을 준다. 다시금 결함 있는 쏘프트웨어공학의 실례를 볼수 있다. 이때 적절한 길이를 검사하도록 하면 보안구멍을 막을수 있다.

3. rlogin바그

Unix 의 **login** 지령의 형식은 다음과 같다.

login[[**-p**][**-h**<host>][**-f**]<user>]

여기서 **-f** 선택은 가입을 《강요》하며 사용자는 통과암호를 넣지 않아도 된다. **rlogin**지령은 사용자들이 원격기계들에 가입할수 있게 한다. 이 지령은 다음과 같은 형식을 가진다.

rlogin[**-l**<user>]<machine>

rlogin데몬은 이 지령의 첫 인수를 취하고 가입요구를 두번째 인수 machine에 보낸다. Linux 와 AIX의 일부 판본들은 이름마당의 문장을 검사하지 않는다. 이때 요구

rlogin -l -froot machine 은 결과적으로

login -froot machine

으로 될것이다. 즉 지정된 기계에 뿌리로서 강제가입된다. rlogin데몬에 의한 적절한 문장론검사는 이 공격을 막을수 있게 한다.

4. Java의 바그

Java애플레트가 어떤 클라스를 실행할것을 요구할 때 Java체계는 클라스이름을 그 안에 있는 점들을 빗선들로 교체함으로써 파일이름으로 변환한다. 즉 클라스 here.it.is는 파일 here\ it\ is에 대응된다. Java체계는 국부디스크에서 이 파일이름을 먼저 탐색한다. 이 탐색이 실패하면 Java는 애플레트에 의하여 제공된 Web봉사기로부터 그 파일을 꺼내오려고 시도한다. 열람기를 지원하기 위하여 국부적으로 설치된 클라스들은 모조클라스로서 취급된다. 따라서 Java는 그러한 클라스를 국부망우에서 검색된 클라스와 혼동하지 말아야 한다.

공격은 다음과 같이 진행될수 있다. 우선 공격자는 피해자의 국부디스크에 들여 보낼 적대코드를 얻어야 한다. 이것은 여러가지 방법으로 얻을수 있다. 명백한 경로는 최근에 접근한 Web페지들의 완충기억기에 열람기가 기억시키는 Web문서처럼 적대코드를 내려 받는것이다(공격자는 어떻게 이 완충기억기가 후에 적대코드로 될수 있게 조직되는가를 알고 있어야 한다).

두번째 단계에서는 사용자가 적대코드를 신용 받는 클라스로 보고 실행하도록 기만하여야 한다.

피해자의 열람기가 클라스 here.it.is를 집행할것을 요구하는 애플레트쓰기는 기만을 하지 않는다. Java는 현재등록부에서 파일 here\ it\ is를 찾을것이다.여기서 그것은 신용 받는 코드로 취급되지 않는다. 공격자는 뿌리에서 시작하는 등록부경로를 가지는 파일이름 례하면 \ here\ it\ is 로 변환하는 클라스이름을 창조하여야 한다. Java는 이 위험을 알고 있으며 따라서 점으로 시작하는 클라스이름을 허용하지 않는다. \ here\ it \ is로 변환되는 클라스이름 .here.it.is는 비법적인 클라스이름으로 인식된다. 공교롭게도 이 규칙은 빗선으로 시작하는 파일이름들을 잡지 못하므로 클라스 \ here.it.is는 신용 받는 코드로 실행될수 있다.

이 문제는 초기의 Java판본에 존재하였는데 쉽게 고쳐 졌다. 파일분리기호들로 시작되는 모든 클라스이름들은 금지된다. Java와 이 바그에 대한 보다 상세한 정보는 [95]에서 볼수 있다.

제4절. 편리한 특징

바그인가 특징인가. 이것은 콤퓨터보안에서 가장 자주 제기되는 질문들중의 하나이다. 유산체계들과 관련한 뒤방향호환성 그리고 설치와 리용의 편리성은 보안을 고려하지 않을 때 체계에 포함시켜야 할 중요한 특징들이다. 만일 이러한 특징들을 알고 있고 또 그것들이 필요 없을 때 제거해 버리는 방법을 알고 있다면 아무런 문제도 없을것이다.

공교롭게도 항상 이렇게 되는것은 아니며 공격자들은 체계바그보다도 이러한 체계특징들을 악용할수 있다.

Unix의 sendmail프로그람이 Internet Worm의 공격대상으로 되여 이 문제에 대한 례증으로 널리 알려 지게 되였다[146]. 우편체계를 설치할 때 체계관리자는 통보문이 목적지에 도착하도록 개설할것을 요구한다. 그러므로 만일 망의 어떤 마디에서의 우편구성이 체계관리자가 그 마디에 가입함이 없이 원격으로 검사되고 수정될수 있다면 편리할것이다. sendmail프로그람은 이 요구를 만족시키는 오유수정선택항목을 가진다. 이 선택항목이 목적지에서 능동으로 설정되면 우편통보문에 있는 사용자이름은 sendmail프로그람이 이 목적체계에서 실행할수 있는 지령모임으로 교체할수 있다. 공격자를 위한 기회는 곧 명백해 졌다.

제5절. 통제된 호출

제5장에서 통제된 호출에 대하여 소개하면서 이러한 프로그람에서의 오유가 보안을 엄중하게 파괴시킬수 있다는것을 강조하였다. 이제 그것을 안받침하는 보다 명백한 립증을 주기로 하자.

1. VMS사용자권한부여기능

디지털(Digital)사의 VAX/VMS조작체계는 사용자들에 대한 접근조종정보를 권한부여파일에 보관한다. 이 파일에 대한 변경은 UAF프로그람을 통하여 진행된다. VMS의 하나의 판본이 다음과 같은 바그로 인하여 피해를 입었다. UAF프로그람은 사용자가 UAF를 실행할 권한이 있는가를 확인해야 한다.

```
Caller: Request Set Authorisation File (parameters)
Systgem: Open Authorisation File;
         Read Caller’s Authorisation;
         If authorized then return(true)
                   else  return(false);
```

공격자는 귀환코드를 무시하고 권한부여파일에 쓰기를 하는데 이때 이 파일은 체계에 의해 닫겨 지지 않는다. 문제는 대단히 민감한 위치에서 발생하는 단순한 프로그람작성오유이다.

2. 가입으로 인한 잠재적인 문제

가입창문을 현시하는 Unix프로그람은 뿌리권한을 가지고 동작한다. 사용자가 가입할 때 가입프로그람은 현재홈등록부를 그 사용자의 홈등록부로 변경시켜 그 사용자의 환경을 설정한다. 그것은 사용자의 **.cshrc**와 **.login**파일들을 읽고 이 파일들에 포함된 지령들을 실행한다. 만일 이 시점에서 가입프로그람이 여전히 뿌리특권을 가지고 동작한다면 사용자는 **.cshrc** 혹은 **.login**과 같은 파일들을 트로이목마(Trojan horses)로 리용할수 있으며 뿌리에 의해서 실행될 지령들을 삽입할수 있다. 그러므로 가입처리의 UID를 사용자에 의해서 정의되였을수 있는 그 어떤 지령을 실행하기전에 사용자의 UID로 설정하여야 한다는것이 본질적이다.

제6절. 우회

론리적접근조종은 론리적체계객체들에 대한 사용자들과 처리들의 접근을 유효하게 한다. 만일 공격자가 론리적접근조종《아래에》 코드를 삽입할수 있다면 이 조종을 우회할수 있다. 또는 공격자는 기억기에 직접 접근하여 론리적접근조종을 우회할수도 있다.

1. AS/400 기계대면부본보기

AS/400(Application System 400)은 중규모콤퓨터들을 위한 IBM조작체계이다. 그것은 재정분야에서 널리 쓰이는 객체지향조작체계로서 상당히 안전하다. AS/400의 보안준위는 QSECURITY체계값을 통해 설정될수 있다. QSECURITY는 10, 20, 30, 40, 50의 값들을 취할수 있다.

- 체계보안준위 10: 보안 없음 ; 이것은 AS/400가 판매될 때 설정하는 기정값이다.
- 체계보안준위 20: 통과암호보안 ; 그러나 객체보안은 없다. 일단 사용자가 가입에 성공하면 그다음은 조종이 없다.
- 체계보안준위 30: 통과암호보안과 객체보안; 앞에서 본 자유접근조종방책들이 객체들에 대한 접근을 막는다.
- 체계보안준위 40: 기계대면부본보기리용에 대한 조종을 첨부한다(아래를 보라).
- 체계보안준위 50: C2 규격준수를 지원한다(그러나 담보하지 않는다).

기술적론의들과 AS/400에 대한 기타 정보들과 조작체계보안관리에 대한 일반적권고들은 [122]에서 볼수 있다.

AS/400에서 기계대면본보기로 씌여 진 기계어프로그람들은 조작체계의 보안조종을 받지 않는다. 기계대면부본보기는 본래 숙련된 AS/400프로그람작성자들이 쏘프트웨어의 성능을 높이는데 리용되였다. 그러나 이 기술을 보안설정을 변경시킬 목적에서 객체들을 림시정정(덧쓰기)하는데 쓸수 있다.

IBM은 우선 기계대면부지령들의 위반을 없애고 기계대면부본보기에서 이러한 지령들을 검출하는 문장론검사기를 리용하여 이러한 변경을 금지시키려고 하였다. 이 문장론검사기는 합법적인 지령들의 표를 참조하였다. 일단 쏘프트웨어작성자들이 그 검사가 어떻게 수행되는가를 알아 내면 그들은 그 IBM표를 모든 지령들을 포함하는 표로 쉽게 바꾸어 버릴수 있다. 조작체계의 완정성을 보호하기 위한 다음 단계로서 AS/400은 사용자상태와 체계상태를 분리하고 사용자령역과 체계령역을 분리하였다. 사용자령역에 상주하는 체계상태프로그람들은 사용자에게 자료기지관리체계와 같은 보호된 쏘프트웨어에 대한 접근을 준다. 보안준위 40에 추가된 이러한 보안기구들은 기계대면부본보기를 허락하는 보안준위 30과는 반대로 동작한다. 그런데도 불구하고 실지는 준위 30이 유산(legacy)쏘프트웨어를 여전히 동작하도록 허락하는 표준구성이다[122].

2. at바그

Unix 지령인 at <time>-f<file>은 사용자들이 후에 지령을 실행할수 있게 한다.그것은 요구된 파일이 지정된 시간에 실행되도록 실패형(spool)등록부인 /user/spool/atjobs 에 넣는다. at프로그람은 그 파일을 실패형등록부에 넣을 때 그것

을 사용자가 읽을수 있는가를 검사하지 않는다. 그러나 실패형등록부에 있는 파일은 at를 실행한 사용자에 의해 읽어 질수 있다. 이 방법을 써서 공격자는 의심스러운 기록통보가 발생되지 않도록 하기 위하여 그것이 실행되기전에

at<time>-f /etc/shadow

를 입력하고 실패형등록부에서 그 일감을 지워 버림으로써 그림자통파암호파일에로의 접근을 얻을수 있다. 이러한 공격은 /user/spool/atjobs를 읽기불가능으로 선언하면 예방할수 있다.

3. 사이드와인더

사이드와인더(sidewinder™)는 Unix체계의 제일 웃부분에 구축된 방화벽(firewall) 제품이다. 사이드와인더는 인터네트에서 방화벽을 통과하여 그뒤에 있는 체계에 침입하는 해커들에 도전하는 방법으로 시험되였다. 이 《시험》등록자리와 사이드와인더에 대한 그밖의 정보는 CIPHER[153]에서 출판되였다.

Unix체계들에서 뿌리는 거의 모든것을 할수 있는 특권을 가진다. 만일 해커가 뿌리로서의 접근권한을 얻을수 있다면 모든 방어수단을 돌파할수 있다. 보안돌파의 영향을 막기 위해 사이드와인더는 체계객체들과 사용자처리들을 서로 다른 령역들에 배치한다. 령역들사이의 분리는 형(type)을 리용하여 실현한다.

매개 령역에는 파일형들과 조작체계호출들의 제한된 모임에로의 접근만이 주어 진다. 매개 령역이 자기의 관리자를 가지므로 특권사용자는 있을수 없다. 공격자가 방화벽체계를 수정할 기회는 두개의 서로 다른 조작체계핵을 사용함으로써 보다 더 제한된다. 사이드와인더가 자기의 Unix 핵심부에서 동작할 때 형강요(enforcement)가 가능하다. 사이드와인더가 관리자적인 핵심부와 함께 기동되면 망접속은 금지된다.

다만 하나의 심중한 해커공격만이 관찰되였다. 해커는 SUID 뿌리 2진코드를 창조하는 부적당한 우편프로그람을 속여 넘김으로써 루트접근을 얻었다. 뿌리와 같이 작업함에도 불구하고 그 이후의 해커의 작용은 체계에 의해 검출되였고 콤퓨터와 사용자와의 대화시간은 결국 끝나버렸다. 해커는 처음때와 같은 구멍을 통하여 빠져 나갔고 그때 현존하는 디스크장치를 가리키는 새로운 장치를 창조하도록 mknod지령을 사용하였다. 이 새로운 장치의 소유자로서 해커는 그 장치에 대한 읽기 및 쓰기특권을 가지며 Unix나 형강요방책에 개의치 않고 그 디스크우에 있는 그 어떤 파일에도 읽기쓰기할수 있었다.

공격은 그이상 더 전진하지 못했다. 다음단계는 다치지 않은 기억내용으로부터 론리파일구조를 재구성하는 도구들을 요구한다. 공격자는 이때 형강요조종기능을 마비시키기 위해 어느 파일을 변경해야 하는가를 알고 있어야 한다. 이 공격은 해커가 미칠수 있는 곳에서 mknod를 제거함으로써(즉 mknod체계호출을 금지된 목록에 배치함으로써) 보호되였다.

4. 지능카드에 대한 공격

지능카드란 집적회로(IC)를 내장한 수지카드이다. 지능카드는 류동하는 사용자들이 행표를 주고 받는 거래를 하는 응용에서 편리하다. 대표적실례를 들면

① 자동출납기계로부터의 현금찾기
② 이동통신: 여기서 사용자들은 자기의 호출에 대해 지불한다.
③ 공공운수: 여기서 사용자들은 자기들의 려행비를 지불한다.

우의 세가지 실례들에서는 사용자권한부여가 필수적이다. 민감한 자료들을 기억하고 있으면서 암호화알고리듬을 수행하는 지능카드는 그 소유자가 장치우에서 물리적조종을 전혀 하지 않으므로 매우 쓸모 있다.

해커는 물리적접근을 얻을수도 있다. 지능카드에 물리적으로 접근할수 있는 사람은 누구나 론리적입력/출력통로들을 리용하지 않고도 다른 방법으로 그것을 취급하고 조종할수 있다[3]. 그러므로 지능카드의 보안해석에서는 공격자가 다음과 같은 문제들을 어떻게 하겠는가를 고려하여야 한다.

- IC의 전원소비를 어떻게 관찰하고 조작할수 있는가?
- 박자신호를 어떻게 조작할수 있는가?
- 계산시간을 어떻게 관찰할수 있는가?
- IC상에서 신호들과 물리적구조들을 어떻게 관찰하고 처리할수 있는가?

공격자는 전원소비나 계산시간을 감시하여 지능카드상에 보관된 비밀열쇠를 추측하는데서 도움을 받을수 있다. 이러한 공격들은 아주 《값 눅은》것들이다.그것은 RAS지능카드처리기들에 대한 시간선도공격에 의해 잘 알려 진것처럼 실행가능하다([47]을 보라). 방어로서는 열쇠의존상태들이 암호화알고리듬의 실행에 반영되지 않는다는것을 확신하는 프로그람작성양식을 리용한다.

IC의 전원공급이나 박자에서의 우연적이거나 계획적인 변화들은 프로그람작성자에 의한 카드상의 처리기를 미리 예견하지 못한 상태로 만들수 있다. 공격자는 일련의 흥미있는 자료(열쇠와 같은)가 착오에 의해 출력되기를 바랄수 있다. 이것 역시 대단히 값눅은 공격이다. 명백한 방어수단은 전원공급이나 박자에 대해서 허용가능한 범위내에 그 신호들의 변화를 유지하는 려파기들이다. IC는 자동적으로 재설정되거나 또는 단순히 카드를 리용하는 응용프로그람에 알리는 기발을 설정하는것으로써 반작용할수 있다. IC제작자들은 장치들에 대하여 담보할수 있도록 이 문제들을 필요한만큼 강조할것이다.

방금 언급된 류형의 조작에 의해 발생되거나 다른 수단들 례하면 방사(radiation)에 의해서 발생된 이행고장들은 암호화함수를 틀린 값으로 돌려 줄수 있게 한다. 미분장애해석에 관한 최근연구들은 동일한 입력에 대한 정확한 결과와 틀린 실행결과들로부터 비밀열쇠를 계산하는것이 가능하다는것을 보여 준다. 이것 역시 결과가 발표되기전에 IC상에 추가적인 검사를 포함시켜 공격을 어렵게 할수 있는 값 눅은 공격수법이다. 보다 값 비싼 수법들은 IC를 조작하는데 레이자를 쓰거나 혹은 계산과정과 기억된 값들을 관찰하거나 IC상에 실현된 비밀알고리듬을 역설계하는데 전자현미경을 리용할수 있다. 이러한 공격들을 완전히 막을수 있다고 생각하는것은 잘못된것이다. 그러나 설계와 제작공정에서는 이러한 공격들이 너무도 비용이 많이 들어 《상업적으로》 수지가 맞지 않도록 할수 있다.

제7절. 결함이 있는 규약의 실현

보안규약들의 추상적인 서술들은 우연수취하기와 같은 해롭지 않는 문장들로 가득차 있다. 그뒤에 숨겨 진 복잡성들은 일단 규약을 실행하려 할 때에만 나타난다. 설계자들은 설계를 쉽게 하기 위해 보안상 결함이 있다는것을 알면서도 그 방안을 택한다. 때로는 설계자들이 그 문제를 즉석에서 발견해 내지 못할수도 있다.

1. TCP권한부여

첫번째 실례는 TCP열기순차에서 리용하는 세가닥응답확인규약(three-way handshake)이다. 보안해석에서는 항상 통보문들이 목적한 수신자들에게 전달되며 통로에서는 관찰될수 없다고 가정한다. 공격자가 취할수 있는 유일한 작용은 자기의 통보문에 가짜송신자주소를 포함시키는것이다. 봉사기 B와의 대화를 열기 위해 의뢰기 A는 다음과 같은 통보문을 보낸다.

A→B:SYN,ISSa.

이 파케트는 SYN(Sychronize Sequence Number)비트묶음을 가지며 32bit의 초기계렬번호 ISSb를 포함한다.

B는 자기의 초기순서렬번호 ISSB와 알려 진 A의 순서렬번호를 보내면서

B→A:SYN,ISSb,ACK(ISSa)

로 응답한다. 실제의 수값 ISSa는 이 통보문에 나타나야 한다.

의뢰기 A는

A→B:ACK(ISSb)

를 보내여 응답확인을 결속한다. 이 통신규약은 긴 초기순서렬번호들이 합리적으로 우연값이여야 안전한다. 따라서 RFC793은 32-bit계수기가 4μs에 한번씩 낮은 자리위치에서 1씩 증가된다고 기술하고있다. 그러나 버클리(Berkeley)유도핵심부들은 매초 128씩 증가하며 매 접속때마다 64씩 증가한다. 여기에는 공격자를 혼돈시키기 위한 우연성이 그다지 많지 않다.

이러한 방법이 일찌기 1985년 로버트모리스(Robert Morris)에 의해서 서술되였고 [105] 후에 스티브 벨로빈(Steve Bellovin)[14]에 의하여 일반화됨으로써 공격이 가능하게 되였다. 공격자 C는 먼저 목표 B에 대한 실제의 접속을 진행하고 순서렬번호 ISSb를 수신한다. 공격자는 다음 A 를 흉내내여 송신자마당에 A의 주소를 가진 파케트를 송신한다.

C(A) →B:SYN,ISSc

B는 이에 대하여 진짜 A에로

B→A:SYN,ISSb',ACK(ISSc)

로써 응답한다. C는 이 통보문을 볼수는 없으나 ISSb'의 값을 추측하고

C(A)→B:ACK(ISSb')

를 송신한다. 만일 추측이 옳다면 사실은 C 가 파케트를 보내고 있는데 B는 이것이 A와의 접속으로 생각한다. C는 이 대화로부터의 출력를 볼수 없으나 봉사기 B우에서 A의 특권을 가지고 지령들을 실행할수 있다.

그러면 B는 의뢰기 A로부터 오는 사용자의 신원과 특권을 어떻게 확인하는가? 만일 사용자가 통과암호를 제공하면서 가입해야 한다면 공격자는 전진할수 없다. 그러나 만일 봉사기가 기계를 신용 받는 호스트로 보고 들어 오는 통보문들에 대한 더이상의 확인을 하지 않으면 공격은 성공할수 있다. Unix에서 신용받는 호스트들은 .rhosts 파일로 선

얻된다. rsh와 같은 규약들은 사용자들이 이미 권한부여된 신용 받는 호스트로부터 오는것으로 가정하는 주소기초권한부여방식을 리용한다. 공격자는 신용 받는 호스트의 IP 주소를 리용하기 위해 이와 같은 규약을 추적할것이다.

수수께끼의 마지막부분은 의뢰기 A이다. 만일 A가 B의 통보문을 받으면 A가 보내지 않은 그 무엇에 대하여 B가 응답하고 있다는것을 알게 될것이며 그 접속을 끊어 버리기 위해 RST 파케트로 응답할것이다. 공격에 간섭하는 A를 견지하기 위해 C는 A가 일련의 리유로 정지되기를 기다리거나 A의 완충기를 동기화요구들로 가득 채워 B로부터 오는 통보문들이 무시되도록 하는 TCP SYN밀물공격을 리용할수 있다.

이 공격을 막기 위해 국부송신자주소밖에서 오는 모든 TCP 파케트들을 막는 방화벽을 리용할수 있다. 이 기구는 모든 신용 받는 호스트들이 국부망내에 있다면 정확히 동작한다. 그러나 신용 받는 호스트들이 밖에도 존재한다면 방화벽은 TCP와 주소기초권한부여방식으로 방화벽을 리용하는 모든 규약들을 막아야 한다. 지어는 주소기초권한부여자체를 포기할수 있다. 그것은 더 큰 재난이 일어 나기를 기다리는것이다. 암호화적인 권한부여방식이 훨씬 바람직하다.

2. Java DNS 바그

Java 의 보안방책은 애플레트들의 작용의 범위를 제한한다. 애플레트는 그가 실행되는 기계우에서 제한된 범위에만 도달할수 있으며 그가 도달할수 있는 다른기계에서도 역시 제한된다. 두번째 제한은 다음규칙에 의해 피복된다[95].

애플레트는 그가 온 봉사기에로 되돌아 가는것을 제외하고는 망접속을 열수 없다.

Java 는 두개의 인터네트봉사기가 같은것인가 아닌가를 어떻게 알아 내는가? 그것은 인터네트이름들을 IP 주소로 변환하는데 령역이름체계(DNS)를 리용한다. 애플레트가 접속을 열 때 Java 체계는

- Web봉사기의 이름을 IP주소들의 목록으로 변환한다.
- 애플레트가 접속하려고 하는 기계의 이름을 IP주소들의 목록으로 변환한다.
- 두개의 목록들에 공통인 항목이 있으면 접속을 허락한다.

이 전략에는 문제점이 있다. 매개 인터네트령역의 소유자는 자기의 령역안에 있는 인터네트이름들을 IP주소로 변환하는것을 담당하고 있다. 악의를 품은 소유자가 자기의 변환목록에 다른 령역으로부터의 IP주소들을 포함시키는것은 막을수 없다. 령역 **attacker.org** 에 적의를 품은 공격자는 비난이 그 어떤 다른 사람에게 떨어 지도록 공격을 개시할수 있다.

실례로 이러한 공격을 례증하기 위해 목적지의 IP 주소를 88.8.8.8 로 하고 **attacker.org** 령역에 있는 Web봉사기는 IP주소 13.13.13.3를 가진다고 하자. 공격자는 자기의 령역안에 있는 DNS 이름 **call.attacker.org** 에 거꾸로 련결을 요구하는 적의를 품은 애플레트를 포함하는 Web페지를 창조한다. 다음 공격자는 아래와 같은 사건들이 일어 나기를 기다린다(그림 8-2).

1. 깨끗한 3부류가 공격자의 Web페지를 보고 그 Web봉사기(IP주소 13.13.13.3)로부터 적의를 품은 애플레트를 내려받기한다.
2. 애플레트가 **call.attacker.org**에로의 접속을 요구할 때 공격자의 DNS봉사기는 DNS이름 **call.attacker.org**의 변환을 요구한다.

3. 공격자는 목록(88.8.8.8,13.13.13.3) 레하면 피해자의 IP주소와 공격자의 Web 봉사기의 IP주소를 돌려 준다.
4. 애플레트로부터 온 Web봉사기의 IP주소가 그 목록에 있기때문에 Java 는 애플레트가 그 목록의 첫번째 IP주소 레하면 피해자의 주소 88.8.8.8에로의 접속을 여는것을 허락한다.

피해자의 위치에서 볼 때 애플레트의 그 어떤 적의를 품은 작용은 깨끗한 3부류에 귀착될것이다. 이 문제는 검사루틴을 변경함으로써 고정된다. 이제는 열람기들이 Web 봉사기의 IP주소를 기억하고 그 주소에로의 접속들만 허락한다.
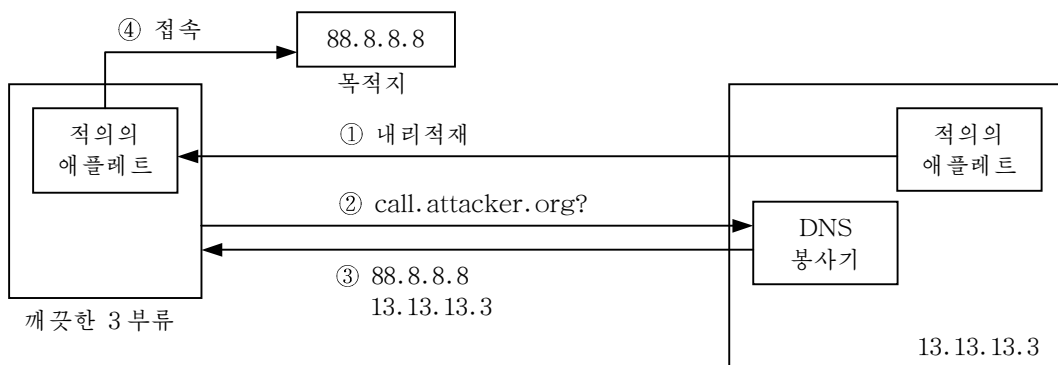
그림 8-2. Jave DNS 바그

제8절. 비루스공격

콤퓨터비루스들은 콤퓨터보안에서 가장 중시해야 할 령역들중의 하나로서 사회적인 관심속에 놓여 왔다. 비루스들은 신문기사들과 영화, 텔레비존방송프로들에 소개되고 있다. 경험이 없는 사용자들에게는 비루스들이 콤퓨터보안에 맞서는 가장 위력한 강적일수도 있다. 그러나 전문가들은 콤퓨터비루스들이 실제적인 피해를 주기는 하지만 지금 떠드는것처럼 과장된 주목을 받을만한 가치는 없으며 훨씬 더 넓은 보안문제의 특정한 한 측면일뿐이라고 보고 있다.

여기서는 《개인용콤퓨터(PC)들과 보안》이라는 화제를 간단히 그리고 일반적으로 보기로 한다. PC들이 처음 시장에 나타났을 때 MS-DOS와 같은 PC조작체계들은 단일사용자용이였고 한 사용자가 기계전체를 조종하였다. 거기에는

- 사용자들을 구분하기 위한 아무런 보안기구도 없었다.
- 《체계》와 《사용자》를 구분하는 아무런 보안기구도 없었다.
- 체계나 사용자파일들의 고의적인 갱신을 막기 위한 아무런 보안기구도 없었다.

개별사용자용이라는것을 고려하면 이 기구들은 요구되지 않으므로 불필요한 특징들을 생략했다고 하여 설계자들을 나무람하지 말아야 한다. 어쨌든 PC의 보급은 새로운 산업을 낳게 하였다. 그것들을 보면 다음과 같다.

- 표계산프로그람(spreadsheet)이나 문서처리기(word processor)와 같은 상업적쏘프트웨어제품들

- 콤퓨터유희들
- PC들의 공동리용 즉 같은 기계를 몇명의 종업원이 함께 리용하거나 PC들을 LAN에 접속한 단체들

이러한 새로운 세계에서 쏘프트웨어는 유연성자기원판에서 교체되게 되였고 직업적쏘프트웨어와 개인적쏘프트웨어가 같은 PC상에서 동작하였으며 회사들의 재산(정보)은 PC들에로 옮겨 져 더이상 중앙 IT부문에 의해 조종되지 않고 개별적사용자들에 의해 조종되게 되였다. 점차 위협적인 각본들이 달라 져 갔고 보안기구의 부족이 스스로 느껴지기(보안의식이 싹트기) 시작하였다.

1. 비루스분류

비루스연구자들은 콤퓨터비루스들을 평가하고 이름 짓고 분류하는 체계를 개발하는것과 콤퓨터비루스들을 다른 형태의 악의 있는 쏘프트웨어와 구별하기 위한 특징들을 정의하는데 많은 노력을 기울였다. CARO규정은 콤퓨터비루스들을 이름 짓고 평가하는데 널리 리용되고 있다. 여기서는 문제를 간단히 하기 위하여 다음과 같이 분류한다.

- 트로이목마(Trojan horse): 이것은 프로그람문서에 서술되지 않은 또 그 프로그람을 실행하는 사용자가 의도하지 않은 은페된 부작용을 가지는 프로그람이다.
- 자체복사비루스: 이것은 어떤 코드에 유효부하를 가지고 덧붙여 진 자체복사코드토막이다. 유효부하는 어떤 통보문을 표시하거나 소리를 내는것과 같은 해를 주지 않는것으로부터 파일을 지우거나 변경시키는것과 같은 유해로운것에 이르기까지의 범위에 놓일수 있다.
- 콤퓨터비루스: 이것은 자기자체를 어떤 프로그람코드에 삽입하여 그 프로그람을 감염시킨다.
- 파도비루스: 감염된 프로그람이 실행될 때에만 작용한다.
- 상주비루스: 이 비루스에 감염된 프로그람이 실행될 때 기억기에 자기를 설치한다. 상주(terminate-stay-resident(TSR))비루스는 지어 감염된 프로그람의 실행이 끝나도 기억기에 그냥 남아 있으면서 다른 프로그람의 실행에로 스스로 이행하여 능동상태로 될수 있다.
- 론리폭탄: 특정한 격동조건이 만족될 때에만 실행되는 프로그람이다.
- 웜(Worm): 감염시키지는 않고 복제하는 프로그람이다.

비루스들에 대한 다음의 분류에서는 공격 받기 쉬운 약점들의 일반적본성을 강조한다. 대부분의 성공적인 공격들은 여러가지 수법들을 결합한것이라는데 대해 간단히 언급한다.

2. PC기동순서

콤퓨터비루스들은 권한 없는 변경에 대한 보호기능이 없는 조작체계들에 있는 공격받기 쉬운 점을 리용한다. 공격 받기 쉬운 범위를 리해하기 위하여 간단히 IBM PC 의 기동순서를 돌이켜 보자. 여기서는 이 공정의 일반적구조에만 흥미를 가지며 ROM이나 유연성자기원판에서의 분구크기나 기억기주소 같은 기술적세부는 무시하기로 한다.

기계를 《시동》시키는것 즉 그의 정상작업방식으로 들어 가게 하기 위해 취해야 할 걸음들을 PC초기기동이라고 한다. 초기에 기억기는 ROM이 없으면 비여 있다. 즉 일단 전원을 끄면 모든 다른 기억기들은 기억내용을 잃어 버린다. PC를 깨끗한 상태로

만들려고 재기동할 때 사실은 물리적효과로 인하여 기억내용들이 즉시에 잃어 지지는 않는다. 그러므로 전원을 끈후 재기동하기전에 약 30초정도 기다려야 한다[49].

조작체계를 기억할 위치만은 ROM과 유연성자기원판이나 하드디스크와 같은 일련의 2차불휘발성기억기이다. ROM은 전체 조작체계를 포함하기에는 너무 작으므로 IBM PC에서는 초기적재(조작체계의 넣기)를 위한 정보가 그림 8-3에서 보여 준것처럼 계층적방식으로 되여 있다.
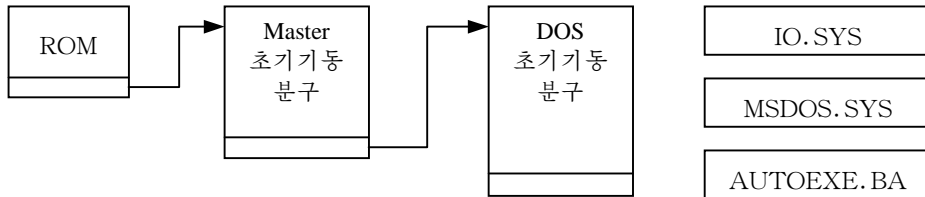
그림 8-3. 전형적인 초기기동순차

- ROM: 2차기억기에 있는 기본초기기동분구를 지적하는 초기화루틴을 포함한다.
- 기본초기기동분구: 유연성자기원판우의 0번 분구, 1번 자리길, 0번 면과 같은 표준위치로서 일부 실행가능한 코드들과 디스크를 구획들로 나누고 초기기동가능한 구획을 가리키는 분구표를 포함하고 있다. 기동가능한 구획들은 기동순서에 따라 순차적으로 탐색된다. 기본초기기동분구는 2차기억기상에 있는 DOS초기기동분구도 지적한다.
- DOS초기기동분구: 실행가능한 코드와 파일배치표(FAT)를 포함한다. FAT는 기억기에서 파일들이 어디에 보관되는가를 기록한다. 거기에 매개 파일에 대한 클라스터(기록들의 그룹)들의 련결목록이 있다. 물리적으로 파괴된 클라스터들은 FAT에 《bad》로 표식된다.
- BIOS(기본입력/출력체계)를 포함하는 IOSYS(또는 IBMBIO.COM)프로그람과 SYSINIT프로그람을 적재한다.
- 조종이 DOS(디스크조작체계)에로 넘어 가며 DOS는 AUTOEXEC.BAT파일을 찾아 보고 지령해석기(보통 COMMAND.COM)를 실행한다. COMMAND.COM 은 사용자의 입력을 재촉한다.

이제는 PC가 사용자들의 입력을 접수하고 응용프로그람들을 실행할수 있게 준비되였다.

3. 초기적재프로그람비루스

공격의 첫번째 대상은 조작체계 그자체이다. 앞에서 본것처럼 PC들에서는 조작체계의 매우 작은 부분만이 ROM에 보관되며 이것은 자동적인 완정성보호를 가진다. 나머지 부분은 감염된 기억매체로부터 읽어 질수 있으며 그것은 이미 앞으로의 공격을 허락하도록 변경되였을수 있다.

PC의 초기기동순서에는 악의를 가진 코드를 삽입할수 있는 충분한 몇가지 가능성이 있다. 초기적재프로그람비루스는 초기기동분구들중의 어느 하나에 상주한다(그림 8-4).

그것은 DOS가 완전한 동작상태로 되기전에 작용을 일으키므로 BIOS기능만을 리용할수 있고 특정한 기계구성방식에 맞게 씌여 진다. 그것이 각이한 기계들에서 다 예견대로 동작하지는 못하지만 대체로 기계를 파손시킬것이다.

스토운드(stoned)비루스(뉴질랜드비루스라고도 한다.)는 기본초기기동분구를 특정한 디스크위치에 복사하고 DOS가 기본초기기동분구로 알고 있는 령역에 비루스코드를 삽입한다. PC가 초기기동할 때 비루스는 정상적인 봉사가 개시되기전에 먼저 실행된다. 브레인(brain)비루스는 DOS초기기동분구에 이 수법을 적용한다. 그것은 본래의 DOS초기기동분구를 기억시킬수 있는 빈 클라스터를 찾아 보고 DOS 초기기동분구가 있던 본래위치에 비루스코드와 재배치된 초기기동분구에로의 지시자를 배치한다. 앞에서와 같이 PC가 초기기동할 때 비루스는 정상봉사가 시작되기전에 실행된다. 비루스는 FAT에서 자기가 리용하는 분구들에 《bad》표식을 함으로써 자기의 존재를 숨길수도 있다.

비루스는 초기기동공정의 뒤단계들에도 역시 자기를 삽입할수 있다. 실제로 IOSYS(Pacman 비루스), COMMAND.COM(Lehigh 비루스), AUTOEXEC.BAT에 대한 《개념의 증명》(proof of concept)이라는 비루스들이 존재한다.
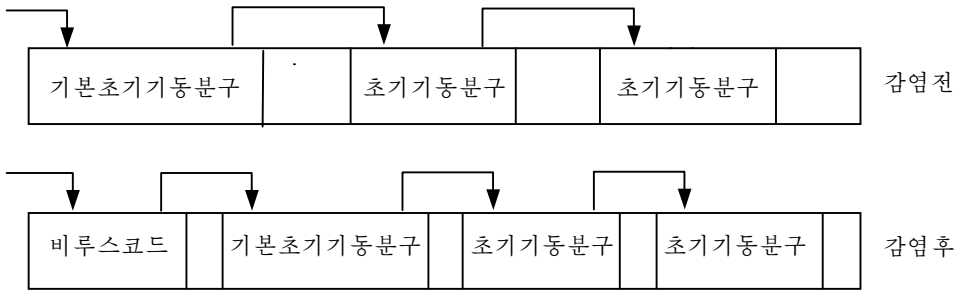
그림 8-4. 초기기동분구비루스의 감염모형

초기기동분구비루스들은 흔히 볼수 있지만 그다음의 비루스들은 보기 힘든것들이다. 초기적재(bootstrap)비루스는 감염된 기억매체 (유연원판)에 있는 체계에 들어 간다. 초기적재비루스를 피하는 효과적인 방도는 깨끗한 디스크로부터 초기기동하는것이다. 그러나 초기기동분구의 일부분이 정적RAM에 보관되여 있으면 초기적재비루스는 거기에 상주할수 있으며 깨끗한 디스크로부터의 기동도 더는 성공을 담보할수 없게 된다.

4. 기생비루스

공격의 두번째 대상은 사용자프로그람들이다. 기생비루스는 .com이나 .exe파일 같은 실행가능한 프로그람에 붙어서 다른 프로그람들을 감염시킨다. 기생비루스는 대표적으로 감염된 프로그람에 자기를 덧붙이고 프로그람의 앞머리에 비루스코드에로의 이행명령을 삽입한다. 비루스의 끝에는 그 프로그람의 시작부에로의 귀환명령이 있다(그림 8-5). 비에나(Vienna)비루스는 감염되는 .com파일들을 이 방식으로 변경시킨다. 그것은 감염된 프로그람의 초기 세바이트를 대피시키고 거기에 그 프로그람에 첨가된 비루스코드의 시작점에로의 이행명령을 배치한다. 프로그람이 실행되면 첫번째 작용은 비루스코드

에로의 이행이므로 비루스코드가 실행되며 다음 대피되였던 본래프로그람의 초기의 세바이트가 복귀되고 실행은 그 프로그람의 시작에로 되돌아 간다.
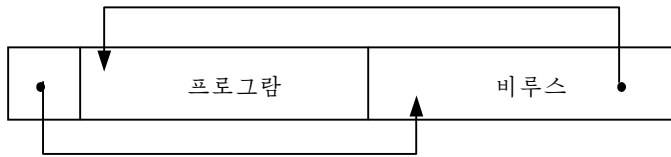
그림 8-5. 기생비루스의 전형적인 감염모형

보다 정교하게 만든 비루스들은보다 작은 쪼각들로 흩어져 감염된 프로그람의 중간에 숨겨 진다. 감염된 프로그람이 실행될 때 비루스가 먼저 실행되고 다음에 그 프로그람이 실행되여 비루스의 효과과 반드시 즉석에서 관찰되지 않도록 한다. 완정성조종이 없는 조작체계에서는 기생비루스가 침습할수 있는 프로그람에 대한 제한이 없다. 보다 강한 조종에 의해 비루스침습을 감염된 프로그람을 실행하는 사용자에게 소속된 파일들로 제한하거나 지어는 프로그람들의 보다 작은 부분모임들로 제한할수 있다.

5. 동반비루스

DOS에서 파일이름들은 《**이름.확장자**》의 형식을 가진다. 대표적인 확장자들로는 **.DIR**, **.COM**, **.EXE** 등이 있다. 추가된 편의기능을 리용하면 사용자들은 실행하려는 프로그람의 완전한 이름을 서술하지 않아도 되며 확장자를 생략할수도 있다. 만일 사용자가 이 수법으로 프로그람을 호출하면 DOS는 먼저 이 이름을 가진 **.COM** 파일을 찾고 다음에 **.EXE** 파일을 찾으며 다음에 **.BAT** 파일을 찾는다. 동반비루스는 이 기정탐색경로를 리용한다.

만일 본래프로그람이 **.EXE** 파일이라면 같은 이름을 가지면서 비루스를 포함하는 **.COM** 파일이 만들어 지고 감염된 프로그람이 실행되게 된다. 이 기술을 AIDS2 비루스가 리용하였다.

그러므로 기정탐색경로들은 편리하지만 위험하다. 같은 류형의 공격이 Unix 에서도 가능한데 **PATH** 환경변수를 따라서 처음으로 탐색되는 등록부에 감염시키려는 목표와 같은 이름을 가지는 감염된 파일을 배치함으로써 실현한다.

6. 마크로비루스

초기기동공정에 삽입되는 비루스는 기계어로 씌여 지며 모든 초기적재프로그람비루스들은 이 준위에서 작용한다. 그러나 이것은 모든 비루스가 기계어로 씌여 져야 한다는 것을 의미하지는 않는다. 일찌기 1989년에 하롤드 하일랜드(Harold Highland)는 표계산프로그람(spreadsheet worksheet)의 자료파일에 자체로 부착하는 마크로비루스의 가능성을 언급하였다［67］. 마크로비루스들은 특별히 흥미를 끌면서 손해를 끼친다.

- 비루스는 자료파일에 붙는다. 그러므로 그것은 실행가능한것 즉 조작체계나 프로그람들을 대상으로 하는 완정성보호기구들을 우회하게 된다.

- 비루스는 고급언어로 씌여 진다. 그러므로 기계어비루스들보다 훨씬 기계의존성이 적다.
- 본문문서들은 전자우편에 의해 널리 교환된다. 이것은 비루스가 확산하는데 매우 좋은 매체로 된다.

마크로비루스들은 1995년에 처음으로 출현하였다. 마이크로소프트의 문서처리체계인 MS-Word는 사용자들이 자기의 문서들이 현시되는 환경을 맞춤하게 설정할수 있게 한다. 형식정보나 기능건들과 아이콘들에 대한 정의는 모두 본문문서와 함께 오는 마크로파일안에 포함된다. 이 본문문서를 열면 마크로안에 있는 명령들이 MS-Word에 의해서 실행된다. 새로운 문서가 만들어 질 때 NORMAL.DOT 파일이 그의 형판(template)으로 리용된다.

마크로를 단순히 자료파일에 붙는 부착물처럼 생각할수 있으나 실제로는 실행가능한 코드의 한부분이다. 그러나 파일을 여는 사용자들은 자기가 프로그람을 실행하고 있다고는 생각지도 못할수 있다. 마크로들을 작성할수 있는 모든 명령들이 마크로파일에 비루스코드를 넣으려 하는 비루스작성자들에게 리용될수 있다. Concept macro-virus 가 바로 그렇게 한다. 그것은 .DOC 와 .DOT 파일들을 감염시킨다. 일단 NORMAL.DOT 가 감염되면 새로 만들어 지는 모든 .DOC 파일은 자동적으로 감염되게 된다.

유연성과 완정성사이에는 이률배반관계가 존재한다. 만일 조작체계전체를 ROM 에 넣는다면 권한이 없는 수정은 불가능하게 되나 반면에 권한을 가진 갱신이나 림시수정 등도 불가능하게 된다.

프로그람파일과 자료파일을 엄격히 구분하는것은 프로그람들이 변하지 않는 환경에서 완정성을 유지하기 위한 좋은 기초이다. 문서처리기는 이러한 프로그람의 좋은 실례이다.

- 자료파일들은 변경할수는 있으나 실행가능한 코드를 포함할수는 없다. 그래서 그것들은 체계를 파괴할수 없다.
- 프로그람파일들은 실행가능한 코드를 포함하며 변경할 필요는 없다. 따라서 모든 프로그람들에 대해서 완정성검사값들을 계산하여 ROM에 기억시켜 두고 어떤 프로그람을 실행하기전에 검사할수 있다.

마크로들은 사용자들에게 보다 유연한 문서처리체계를 제공하기 위해 도입되였다. 그러나 마크로들은 자료와 프로그람사이의 구별을 모호하게 하므로 새로운 보안문제를 초래한다.

체계에서의 변화는 새로운 공격 받기 쉬운 점을 만들어 놓았다.

7. 새치기의 방향바꾸기

비루스는 특권방식에서 실행될 때 가장 큰 효과를 낸다. 극소형처리기의 특권방식에로 들어 가기 위해서는 새치기(중단)가 발생되여야 한다. 그러면 조작체계는 새치기표로

부터 새치기처리기의 주소를 찾는다. 따라서 새치기표 또는 그와 류사한 구조가 주되는 공격목표로 된다. 새치기처리기의 주소를 변경함으로써 조작체계가 비루스를 실행하도록 방향을 바꿀수 있다. 비루스는 스스로 기억기TSR(terminated and-stay-resident)프로그람처럼 상주하게 하며 대응하는 새치기가 발생할 때마다 실행된다.

이 공격은 새치기처리기를 변경시키지 않으며 또 새치기처리기를 검사하는 완정성조종기구도 이 공격을 막을수 없으므로 매우 효과적이다. 류사한 공격은 파일배치표(FAT)에 있는 항목들을 변경시키는 방법으로도 진행할수 있다. FAT의 항목들은 비루스를 지적하도록 변경되며 계속하여 본래의 파일에로의 련결을 포함한다.

이 형태의 비루스를 제거하면 또 다른 파괴를 일으키게 된다. 비루스는 표와 파일사이의 련결의 일부분이다. 만일 비루스가 제거되면 련결이 파괴되고 파일은 회복될수 없으며 결국 그 파일은 조작체계에로 돌아 올수 없다.

8. 위장

비루스는 각종 수단에 의한 검출을 피하려고 한다. 그것은 감염된 프로그람을 압축하여 감염이 기억기의 리용을 증가시키지 않도록 할수 있다. 스텔스(stealth)비루스는 FAT에 《bad 》로 표시된 분구에 숨어 있다. 이리하여 다른 프로그람들은 정상적으로 디스크를 읽을 때 이 분구를 뛰여 넘는다. 이 비루스는 자기의 존재 례하면 감염시킨 파일의 길이, 날자, 검사합 등을 검출하기 위한 새치기들을 가로 채고 감염전의 값들을 돌려 준다. 다형성(polymorphic)비루스는 자신을 암호화하며 또한 패턴인식스캐너들에 의한 검출을 피하기 위해 감염시킬 때마다 새로운 암호열쇠를 리용한다. 말티-파타이트(multi-partite)비루스는 검출을 보다 어렵게 하기 위해 여러가지 감염형태들을 조합한다. 슬로우 인펙션(slow infection)비루스들은 즉시적인 검출을 피하기 위해 감염의 속도를 조종한다.

제9절. 항비루스쏘프트웨어

비루스공격들은 완정성조종의 부족점을 리용한다. 따라서 자신을 방어하기 위해서는 그러한 조종들을 추가하여야 한다. 비루스전용인 일부 유용한 보호기구들이 있지만 대체로 그것들은 일반적으로 완정성을 전제로 한다. 방어전략은 다음과 같은 요소들을 가진다.

- 예방: 비루스가 체계를 감염시키는것을 막는다.
- 검출: 체계를 감염시킨 비루스를 검출한다.
- 반작용: 체계를 깨끗한 상태로 회복시킨다.

성과적인 비루스보호를 위해서는 관리수단들과 사용자의식이 필수적이다.

1. 물리적이며 행정적인 조종

물리적이며 행정적인 조종들은 체계에로 들어 오는 비루스를 막기 위한 좋은 수법이다. 이러한 수단들중 일부는 아주 단순하다. 만일 플로피디스크에 쓰기를 진행하지 않으면 쓰기보호딱지를 붙여 비루스가 그것을 감염시킬수 없게 한다. 만일 조작체계가 접근

조종을 제공한다면 그것을 적절히 리용할수 있다. 실례로 망봉사기들에 있는 모든 응용프로그람들에 대한 파일허가를 읽기와 실행만으로 설정한다.

비루스가 체계로 들어 올수 있는 곳에 조종점들을 배치한다. 비루스제거프로그람이 설치되여 있는 검역기계에서 모든 새로운 프로그람들을 검사한다. 이때 될수록 낮은 특권을 가진것부터 례하면 Guest(손님)으로부터 검사한다. 더 좋기는 비루스스캐너를 동작시키는데 관문기계를 리용하여 조직에로 들어 오는 모든 플로피디스크들을 검사한다.

만일 플로피디스크가 깨끗하다고 인정되면 그 디스크에 표식을 붙이고 내적인 리용을 위한 검열을 끝낸다. 만일 표식이 디스크우에 붙이는것이라면 그후에 조직내에서 권한이 없는 디스크들을 적발해 내는것은 사용자의식에 달려 있다. 만일 디스크에 전자적인 표식이 씌여 진다면 조직내에 있는 기계들은 그의 존재를 검사하고 표식이 없는 디스크들을 거절할수 있다. 최근에는 방화벽제품들에 흔히 망을 통해서 들어 오는 비루스들을 찾아 내기 위한 비루스검사기가 장비되고 있다.

정기적으로 비루스검사를 진행하며 비루스제거프로그람을 항상 최신의것으로 보유해야 한다. 비루스제거프로그람은 매개 사용자의 가입각본에 포함될수 있다. 체계편의프로그람들은 미리 정해 진 시간에 자동적으로 검사를 수행할수 있다.

실례로 Unix체계에서 관리자는 **cron**편의프로그람에 어느 때 완정성검사프로그람을 동작시키라고 지시할수 있다. 한개의 보호기구에만 의존하지 말고 여러가지 기술들을 조합하여 리용해야 한다.

유사시에 비루스공격에 어떻게 대응하겠는가를 구체적으로 계획하여야 한다. 비루스공격에 대한 서투른 반작용이 비루스 그자체보다 더 큰 피해를 초래한다는것이 자주 지적되고 있다. 공격후에 체계를 정확히 복구하려면 깨끗한 여벌을 가지고 있는것이 기본이다. 그러나 비루스가 검출될 때에는 그것이 이미 보관된 모든 여벌들에 영향을 미쳤다고 보는것이 옳다.

2. 암호학적검사합

암호학적검사합들은 표준완정성보호기술이다. 검사합은 보호하려는 파일의 깨끗한 판본에 대해서 계산된다. 이 검사합은 안전한 장소 례하면 ROM 이나 CD에 보관된다. 이 파일이 리용될 때마다 그의 현재 판본에 대하여 계산된 검사합이 보관된 검사합과 비교된다. 이리하여 원본에 대한 그 어떤 변경도 검출할수 있다. 검사합계수기는 비루스의 존재를 검출하기 위하여 그 비루스에 대하여 알지 못해도 된다.

검사합계수기들은 검사합이 재계산되여야 할 때 즉 파일이 변경될 때나 깨끗한 검사합들이 잃어 졌을 때 공격 받기 쉽다. 그러므로 그것들은 쏘프트웨어를 개발하는 조직들보다 고정된 쏘프트웨어도구묶음을 리용만 하는 환경에 적합하다. 검사합계수기들은 감염을 일으킨 비루스를 지적하지 못하며 일단 감염을 검출한 다음 그이상의 작용을 계획하는것은 보다 어려운 일이다.

보안체계는 자기가 보호하려는 객체의 형태를 알면 공격자의 형태를 알지 못해도 된다. 보안체계는 객체들의 변경을 검사할수 있다.

3. 스캐너

스캐너(scanners)들은 콤퓨터비루스들의 알려 진 모양들(비루스특징)을 가지고 파일들을 탐색한다. 즉 검출할 비루스에 대하여 알고 있어야 하며 따라서 부단한 갱신을 요구한다. 스캐너들은 대부분이 대중적인 비루스제거프로그람이다. 검사합계수기들은 검사합이 얻어 진후에야만 효과를 가지지만 스캐너들은 아무런 준비없이 리용될수 있으며 또 사용자에게 무슨 비루스가 발견되였는가를 알려 주어 퇴치할수 있게 해준다.

한편 스캐너들은 능률적인 고속탐색기술들을 요구하므로 파일의 시작과 끝만을 검사할수 있으며 따라서 말티-파타이트(multi-partite)비루스는 쉽게 이 그물을 벗어 날수 있다. 전통적으로 스캐너들은 비루스검사를 진행할 때 기억기에 있는 모든 파일들을 검사하였다. 접근스캐너(on-access scanner)들은 파일에 대한 접근이 요구될 때에만 검사한다.

스캐너들은 또한 비루스서고들에 있는 콤퓨터비루스들의 수가 한정없이 증가한다는 사실을 고려해야 한다. 변이엔진(mutation engines)은 새로운 비루스변종들을 자동적으로 생성한다. 다형질비루스들은 하나의 비루스종류가 많은 서로 다른 모양을 나타내도록 모습을 부단히 변경시킴으로써 패턴인식기술들을 쓸수 없게 한다. 간단한 패턴탐색스캐너들은 이 문제에 대처하기가 불가능하다. 마크로비루스들은 문자렬대조방법으로는 비루스코드를 검색할수 없게 고급언어로 작성되므로 비루스스캐너들에게 새로운 문제를 제기한다.

원래 문자렬대조의 시도는 악의를 가진 쏘프트웨어를 검출하자는것으로서 코드의 문장론적속성들을 리용하므로 실제로는 요구되는 보호에 가까울뿐이다. 따라서 개선된 스캐너들은 비트렬패턴대신에 인식패턴 즉 의미론적인 속성들을 인식하는 방향으로 움직이고 있다.

끝으로 틀린 긍정(false positive)의 문제가 있다. 파일은 우연히 비루스와 같은 특징(signature)을 포함할수 있는데 스캐너는 경고를 내게 된다. 스캐너가 자기의 자료기지에 더 많은 비루스특징들을 가지고 있을수록 이런 현상은 더 자주 일어 날것이다.

객체들은 합법적으로 변경될수 있으므로 보안체계는 보호할 객체의 모양을 알고 있든가 아니면 공격자의 모양 혹은 그의 행위를 알고 있어야 한다. 그래야만 공격자들의 존재를 검사할수 있다.

이 장의 문헌안내

문헌 [85]와 다음의 Web페지에서는 콤퓨터프로그람보안의 결함들에 대한 분류를 볼수 있다.

http://www.itd.nrl.navy.mil/ITD/5540/main_fra.html.

다음주소에는 보안바그들을 위한 우편목록(mailing list)인 Bugtraq가 보관되여 있다.

http://www.netspace.org/lsv-archive/bugtraq.html.

문헌［142］에서는 Intel의 80x86처리기의 보안결함들을 해석하고 있다.

콤퓨터비루스에 대한 많은 책들은 여러가지 변종들의 라렬에 지나지 않는다.이 점에서 ［8］과 ［49］가 주목할만한 례외들이며 아주 상세한 기술정보들을 주고 있다. 만일 우에서 언급한 실제공격들에 대한 이야기에 흥미가 있다면 ［148,140,43］을 보시오.

이것들은 일반독자들을 위해 씌여 진 공격들에 대한 사실적이야기들이다. 1989년 6월판 Communications of the ACM(Vol.32,no.6) 과 ［44］에서 Internet Worm에 대한 자료를 볼수 있다. 문헌［22］에는 마크로비루스에 대한 방대한 론의가 주어 진다. ［95］에는 Java의 바그에 대한 상세한 정보가 있다. CERT보고서나 비루스전문뉴스그룹에 신청하여 새로운 공격들에 대하여 학습할수 있다.

련습문제

1. 임의의 길이를 가지는 문자렬을 입구하여 특수한 NULL문자에 의해서 문자렬의 끝을 검출하는 문자렬조작에서 완충기자리넘침문제들이 많이 발생한다. 이 NULL기호를 리용하여 어떤 문자렬조작이 탄창에 보관된 귀환주소를 고쳐 쓰는것을 검출하는 체계를 설계하시오.
2. 안전한 기동(secure boot)은 중요한 하나의 보안기구이다. 안전한 체계기동을 제공하는 리용가능한 제품들의 봉사를 유도하시오. 이러한 기구를 어떻게 실현하겠는가?
3. 벨-라파둘라모형의 위임 및 자유보안방책으로써 비루스의 감염을 방지하거나 줄일수 있는가?
4. 어떤 환경에서 비루스가 쓰기보호된 파일을 감염시킬수 있는가?
5. 비루스를 적발해 내기 위한 검사합과 스캐너의 우점과 결함은 무엇인가?
6. 마크로비루스들은 콤퓨터비루스에 의한 위험을 더 한층 새롭게 증가시켰다. 항비루스프로그람이 이 새로운 비루스에 어떻게 대응할수 있는가?
7. 쏘프트웨어는 실행가능한 파일을 암호화하며 보호할수 있다. 프로그람을 기동하면 실행가능한 파일의 암호가 해득되고 실행된다. 필요한 암호열쇠를 보관하는 세가지 가능성들을 고찰하자.

   ○ 열쇠를 콤퓨터의 어느 곳엔가 숨겨 놓는다.
   ○ 열쇠를 지능카드와 같이 사용자가 보관하는 통표(token)안에 보관된다.
   ○ 열쇠를 암호를 해득하고 명령을 실행하는 수동조작할수 없는 장치에 보관한다.

   이 세가지 방법들의 보안효과를 비교하시오.
   공격자가 이 방법들에 의한 조종을 어떻게 우회할수 있는가?
8. 자신의 사업에 적용된 콤퓨터악용에 관한 법률과 규정에 대하여 소론문을 쓰시오.

제9장. 보안평가

안전한 체계의 사용자들은 그들이 사용하는 제품들이 적절한 보안을 제공한다는 몇가지 종류의 담보를 필요로 한다. 그들은

1. 제작회사 /봉사제공자의 말에 의거할수 있다.
2. 체계를 자체로 시험할수 있다.
3. 제3자에 의한 공정한 평가에 의거할수 있다.

사용자들이 우의 두번째를 선택할수 있으려면 보안전문가가 되여야 한다. 그러나 대부분의 사용자들은 이러한 수준에 도달하지 못하고 있다. 그래서 어떤 보안평가방법은 신용 받는 보안제품을 선택하는 길밖에 없다. 이 장은 보안평가방법들을 개괄하고 현재의 평가방법들이 어떤 편리성을 제공하는가를 론의한다.

목적

- 근본문제가 임의의 평가기준모임을 찾는것이라는것을 옳게 인식하는것이다.
- 평가기준들을 비교하기 위한 기준을 제기하는것이다.
- 주요한 평가기준에 대한 견해를 주는것이다.
- 평가된 제품과 체계의 우점들을 조사하는것이다.

제1절. 소개

신용콤퓨터보안평가기준(TSSEC, 오렌지부크)[112]은 널리 리용할수 있는 첫 평가기준이였다. 많은 보안체계의 제작자들은 아직 자기들의 제품들에 오렌지부크의 평가를 인용하고 있다. 오렌지부크의 보안기준을 개선하기 위하여 수많은 기준들이 개발되였으며 제기된 여러가지 평가기준들이 단일화되였다. 새로운 평가기준들은 보통 오렌지부크를 어떻게 자기들의 골격에 맞추는가를 설명하는듯한 느낌을 주고 있다. 중요한 문서들은 다음과 같다.

- 정보기술보안평가기준(Information Technology Security Evalution Critena-ITSEC)[117]
- 카나다신용콤퓨터제품평가기준(Canadian Trusted Computer Product Evaluation Criteria)[150]
- 련방기준(Federal Criteria)[115]
- 공통기준(Common Criteria)[26]

여기서는 다음과 같은 물음들을 제기하고 보안평가에 대한 론의를 진행하기로 한다.

평가의 목표는 무엇인가?

평가기준은 여러가지 다양한 응용들에 리용되며 일반적인 보안요구들을 만족시켜야 하는 조작체계와 같은 제품들이나 또는 주어 진 응용의 요구에 맞게 조립된 제품들의 모임인 체계들을 대상으로 한다. 첫 경우에는 일반적인 요구들을 수용한 모임을 찾아야 한

다. 두번째 경우에는 요구파악과 분석이 각각 독립적인 평가부분들로 된다. ITSEC는 체계의 평가에 적합하다.

체계와 제품사이의 차이는 보안평가에서의 근본난점을 강조하고 있다. 사용자들은 보안전문가는 아니지만 고유한 보안요구들을 가지고 있다. 전형적요구들을 수용한 일반기준에 관계되는 기성제품들의 평가는 비전문가에게 쓸모 있는 판단기준으로는 될수 있지만 제품들이 실제의 보안요구들을 만족시키지 못할수도 있다. 주문체계의 평가는 파악된 요구들을 처리한다. 그러나 이 평가는 비전문가인 사용자에게 보안요구들을 적절하게 파악할것을 요구한다. 앞으로는 일반대중을 대상으로 한 보안평가와 특수한 의뢰자를 대상하는 보안상담과제사이의 경계가 없어 지게 될것이다.

평가의 목적은 무엇인가?

오렌지부크는 다음의 항목들을 구별한다.

- **평가**(Evaluation): 제품이 요구된 보안속성들을 가지고 있는가를 평가한다.
- **보증**(Certification):평가된 제품이 주어 진 응용에 적합한가를 평가한다.
- **인정**(Accreditation):증명된 제품을 주어 진 응용에 리용하기로 결정한다.

이 항목들은 오렌지부크의 용어들이다. 물론 다른 참고서들에서는 이와 다른 용어를 사용하든가 또는 같은 용어를 다르게 사용할수도 있다. 그러나 각이한 작용들에 붙여 진 이름들은 그것들이 의미하는 각각의 목적들에서의 근본차이보다 중요치 않다.

평가방법은 무엇인가?

평가방법은 다음의 두가지 결과를 초래하지 말아야 한다.

1. 평가된 제품이 심각한 결함을 포함하고 있다는것을 뒤늦게야 발견한다.
2. 같은 제품에 대한 서로 다른 평가결과가 일치하지 않는다. 때문에 반복성과 재생성은 평가방법이 자주 부닥치게 되는 요구들이다.

보안평가는 제품지향 또는 공정지향으로 될수 있다. 제품지향방법은 제품을 조사하고 검사한다. 이 방법들에서는 공정지향방법보다 제품에 대하여 더 많은 정보가 알려 지지만 여러가지 평가결과가 서로 다르게 나타날수 있다.

공정지향(검사)방법들은 문서와 제품개발공정을 조사한다. 그것들은 값 눅고 반복할수 있는 결과들을 얻기가 훨씬 쉽지만 그 결과자체는 크게 가치가 없을수 있다. 유럽정보기술보안평가지도서 [118] 의 첫 판본은 내용을 압도하는 반복성의 생동한 실례이다.

평가기준의 구조는 무엇인가?

보안평가는 제품/체계가 안전하다는 담보를 목적으로 한다. 보안과 담보는 다음과 같은 항목들에서 관계될수 있다.

- **기능성**(Functionality): DAC, MAC, 인증, 검사와 같은 체계의 보안특징들.
- **효과성**(Effectiveness): 리용된 기구들이 주어 진 보안요구에 적합한가?
- **담보**(Assurance): 평가의 철저성.

오렌지부크는 표준적인 DoD 요구들의 주어 진 모임을 위한 평가클라스들을 정의한다. 때문에 그의 평가클라스들의 정의에서는 이 세가지의 측면들이 동시에 고찰된다. ITSEC는 새로운 보안요구들을 취급할수 있는 유연한 평가구조를 제공한다. 따라서 우의 세가지 측면들은 독립적으로 취급된다.

평가공정의 조직적구조는 무엇인가?

보안제품의 속성들에 대한 보안평가는 독립적이고 널리 인정된 판정기준에 도달하여야 한다. 독립적인 평가기구는 정부기관일수도 있고 적절히 인정된 개별기관[135]일수도 있다. 두가지 기구들에서 다 정부단체가 평가공정을 책임지고 증명서들을 발행한다. 인정된 평가기구들이 자기자체로 증명서들을 발행하거나 평가자들의 전문기술대신에 경험적인 증언이 사실상 공식적인 믿음으로 바뀔수도 있다.

만일 모든 평가들이 하나의 정부기관에서 진행된다면 평가의 일관성을 담보하기 위한 조직적부가기구가 필요 없다. 그러나 시간이 지남에 따라 해석에서의 편차(해석표류)가 생겨 날 위험성이 있다. 평가기구의 경쟁부족과 자원의 제한으로 인하여 평가속도가 느려 질수도 있다. 또한 숙련된 평가자가 다른 직무로 이동할 때 직원이동의 문제도 있을수 있다.

정부기관은 평가를 유상으로 할수도 있고 혹은 무료공공봉사로 할수도 있다.

개별평가기구들을 가진 환경에서는 인증국들이 각이한 기구들사이의 평가의 일관성(반복성과 재생성)을 보장하도록 해야 한다. 해석을 제멋대로 하지 않도록 평가기준들을 정확히 형식화하는것이 보다 중요하다. 한편 부당한 압력에 의해 부정확한 평가결과가 얻어 지지 않도록 사전대책을 취해야 한다.

기타 조직적인 측면들은 평가의 인증자와 제품생산자 그리고 평가기구사이의 계약상의 관계에 관련된다. 그리고 평가증명서를 발급하기 위해서와 평가된 제품의 변경에 대한 재평가를 위해 평가의 시작에 대한 적절한 절차들도 있어야 한다.

평가의 비용과 리익은 무엇인가?

평가를 위한 비용외에도 평가에 필요한 증거들을 얻는데와 평가자들을 키우고 평가팀과의 련계를 취하는데 소비된 시간과 같은 간접적인 비용들도 고려해야 한다. 평가의 비용을 고려할 때 기성제품과 주문제품의 평가를 구별해야 한다. 첫번째 경우(기성제품인 경우)에는 평가의 담보자가 많은 사용자들에게 비용을 전개시킬수 있다. 그러나 두번째 경우(주문제품인 경우)에는 평가담보자와 한명의 주문자가 모든 비용을 부담하게 될수도 있다.

제2절. 오렌지부크

보안평가지침을 만들기 위한 사업은 1967년에 시작되여 보안제품들을 평가하기 위한 첫번째 평가지침인 신용콤퓨터보안기준오렌지부크(Orange Book)[112]를 만들어 냈다. 오렌지부크의 저자들은 보다 일반적인 응용에 쓸수 있는 문서를 만들기 위해 다음과 같은것들을 제공하려고 하였다.

- 사용자가 콤퓨터보안체계에 적용할수 있는 믿음의 정도를 평가하기 위한 기준
- 콤퓨터보안체계들의 제작을 위한 지침
- 콤퓨터보안체계들을 도입할 때 보안요구를 묘사하기 위한 기초

보안평가는 체계의 보안과 관련되는 부분 즉 다시 말하여 신용계산기초TCB(제5장 2절)를 검토한다. 오렌지부크의 접근조종방법은 벨-라파둘라모형(제4장 2절) 즉 보안표식들의 살창에 기초한 자유 및 위임접근조종으로서 이미 보았다. 참조감시기는 주동체들이 객체들에 접근하려 할 때 권한을 가지고 있는가를 확인한다.

보다 강한 담보는 형식적인 방법들과 간단한 TCB들 그리고 구조화설계방법들과 관계된다. 벨-라파둘라는 오렌지부크보안방책에 따르는 형식적인 모형의 하나이지만 다른 모형들은 TCSEC 평가에도 리용할수 있다. TCB를 보다 간단하게 할수록 보다 포괄적인 해석을 가능하게 할것이다. 때문에 복잡한 체계들은 보다 더 낮은 평가클라스로 떨구어야 한다.

1. 보안 및 평가범주

오렌지부크의 평가클라스는 보안요구의 표준형태를 취급하도록 설계되였다. 이 리유로 하여 특정한 보안특징요구와 담보요구가 평가클라스의 정의안에서 결합된다. 평가클라스의 서술에서 주요표제들은 다음과 같다.

- **보안방책**(Security Policy): 주동체와 객체의 의미로 표현한 자유 및 위임접근조종.
- **객체의 표식달기**(Marking of objects): 표식들은 객체의 중요성을 나타낸다.
- **주동체의 식별**(Identification of subject): 개별적인 주동체들을 식별하고 인증하여야 한다.
- **책임추적가능성**(Accountability): 보안관련사건들의 검열기록들을 보존하여야 한다.
- **담보**(Assurance): 조작담보는 주로 보안구성방식에 귀착된다. 수명담보는 시험과 구성관리, 설계방법들과 같은 문제들에 귀착된다.
- **문서화**(Documentation): 체계관리자들과 보안체계의 사용자들은 그의 보안특징들을 적절하게 설정하고 사용하기 위한 지도서를 요구한다. 평가자들은 시험과 설계문건을 요구한다.
- **일관한 보호**(Continuous protection): 보안기구들은 함부로 변경해서는 안된다.

오렌지부크는 4개의 보안구분과 7개의 보안클라스들을 정의하는데 이러한 기준들을 리용한다. 더 높은 준위 보안클라스에 있는 제품들일수록 더 많은 보안기구들을 제공하며 담보가 잘될수록 보다 더 엄격한 분석을 제공한다. 4개의 구분들은 다음과 같다.

D 최소보호
C 자유보호
B 위임보호
A 검증된 보호

오렌지부크의 보안클라스들은 증가하는 순서로 정의되였다. 한 클라스의 모든 요구들은 보다 더 높은 준위 모든 클라스들의 요구에 포함된다. 오렌지부크는 국가적인 보안조직에 의하여 집행되는 응용분야에 의존하지 않는 보안평가를 위한 기초이다.

D-최소보호

이 클라스안에서 평가를 하려는 제품을 찾게 되지만 그것은 오렌지부크의 어떤 클라스의 요구와도 맞지 않는다.

C1-자유보호

C1체계들은 협동하는 사용자들이 동일한 완정성준위에서 자료를 처리하는 환경을 요구한다. 개별적인 사용자들 및 그룹들에 기초한 자유접근조종은 사용자들이 한 조종방

식안에서 객체들에 대한 접근을 공유할수 있게 한다. 사용자들은 자기자신을 증명해야 하며 자기들의 신분을 인증 받아야 한다.

TCB는 동작담보를 위하여 자기자체의 실행령역을 가져야 하며 정기적으로 TCB의 정확한 동작을 검사하는 특징들이 반드시 있어야 한다. 생활주기담보는 《명백한 결함》을 시험하는 보안에 귀착된다. 사용자지도서와 시험문서, 설계문서들이 제공되여야 한다. C1체계를 개괄한다면 편리한 환경으로서는 적합하지만 강력한 보안을 제공하지는 못한다는것이다.

C2-조종된 접근보호

C2체계들은 사용자들이 자기의 행동을 개별적으로 책임지게 한다. 자유접근조종은 단일사용자단위로 실시된다. 이때 접근권한들의 전파를 통제해야 한다. 주동체는 선행한 주동체가 생성한 정보를 포함하는 TCB가 취급한 객체에 접근하지 말아야 한다. C2클라스의 정의에 명백히 서술된것처럼 보안관련사건들의 검사리력을 보관하여야 한다.

시험과 문서화는 새롭게 추가된 보안특징들을 포함해야 하지만 담보는 아직 적다.

일반적으로 C2는 본질적으로 연약함에도 불구하고 상업적인 응용에서는 가장 합리적인 클라스로 간주된다[7]. 대부분의 주요제작자들은 C2로 평가된 조작체계나 자료기지관리체계의 판본들을 제공한다. 때때로 이들은 C2에 맞는 구성의 체계설치를 돕는 전용의 봉사프로그람들을 제공한다[122,63].

B1-표식 붙은 보안보호

구분(Division) B는 클라스화된 자료들을 처리하며 위임된 벨-라파둘라방책들을 실행하는 제품들을 목적한것이다. 여기에는 계층적인 분류준위와 비계층적인 클라스로 구축된 매 주동체와 객체를 위한 표식들이 있다. 이 표식들의 완정성들을 보호해야 한다. 식별과 인증은 주동체의 보안표식을 결정하는데 쓰인다.

보호가 표식에 의존한다면 그것이 다른 체계나 또는 인쇄기에로 반출될 때 표식된 객체에 무슨 일이 생겼는가를 주시해야 한다. 해결책은 반출통로의 속성에 의존한다. 통신과 I/O통로들은 단일준위 또는 다중준위일수 있다. 다중준위통로들에서 객체들은 자기의 표식을 가지고 반출된다. 단일준위통로에서는 TCB가 인증된 사용자와 반출된 정보의 중요성준위를 지적한다. 사람이 읽을수 있는 출력도 역시 표식을 붙여야 한다. 례를 들면 중요한 문서의 매 페지에는 분류를 인쇄하여야 한다.

보다 높은 준위 담보을 얻자면 보안방책의 비형식적인 또는 형식적인 모형이 요구된다.이때는 검사와 문서화가 보다 철저하게 진행되여야 한다. 설계문서, 원천코드, 목적코드를 분석하여야 한다. 검사에서 검출되지 않은 모든 결함들을 제거해야 한다.

그러나 클라스 B1는 TCB의 구조에 관해서 반드시 요구되는것은 아니다. 여기서 다중준위안전Unix체계 또는 자료기지관리체계와 같은 복잡한 쏘프트웨어들은 B1클라스의 인정을 받아야 한다. B1클라스는 구획(compartment)이 있는 체계의 고준위환경을 목적한것이다.

B2-구조화된 보호

B2클라스는 주로 체계의 설계에 대한 요구를 추가함으로써 담보를 증가시킨다.위임접근조종은 물리적인 장치들에로의 접근도 관리한다. 사용자들은 자기들의 보안준위의 변경에 대하여 알아야 한다. 가입과 초기인증을 위한 신용 받는 경로가 있어야 한다.

보안방책의 형식화모형과 체계의 서술적인 고준위설계묘사(DTLS)가 요구된다. 모듈화는 체계방식의 중요한 설계특징이다. TCB는 처리들을 격리시키기 위한 명백한 주소공간을 제공한다. 하드웨어는 토막화와 같은 기억기관리를 제공한다. 잠복통로분석을 진행해야 하며 잠복통로를 만드는 사건들을 검사해야 한다. 보안검사는 TCB가 침투하기 힘들게 되여 있다는것을 확증한다.

B3-보안령역

B3체계들은 침투하기 매우 어렵게 갱신되였다. B3클라스안에서 많은 새로운 요소들이 보안관리를 한다. 보안관리자가 지원된다. 기구들의 검열은 보안관계사건들의 출현과 축적을 감시하고 수상한 정황에서는 자동적으로 경고를 낸다. 체계실패후에는 믿음성 있는 회복을 진행해야 한다.

TCB의 복잡성을 최소화하고 보안과 관계되지 않는 모듈들을 배제하는데 보다 많은 체계공학적노력들이 돌려 진다.

A1-검증된 설계

A1클라스는 B3과 기능적으로 동등하며 형식적인 방법을 리용하여 가장 높은 준위의 담보를 준다. 방책과 체계의 형식적묘사와 일관성증명들은 TCB가 정확히 실현된다는 높은 급의 담보를 보여 준다. A1클라스를 위한 평가에서 나서는 요구들은 다음과 같다.

- 보안방책의 형식화모형,
- TCB기능의 추상적정의를 포함하는 가장 높은 준위의 형식적묘사(FTLS),
- 모형과 FTLS사이의 일관성증명(가능한 한 형식적인),
- TCB실현이 FTLS와 일치하다는것을 보여 주어야 한다.
- 잠복통로의 형식적분석(시간조종통로들은 비공식적으로): 잠복통로들이 계속 존재하면 바로 잡아야 하며 대역폭을 제한해야 할수 있다.

더우기 보다 엄격한 구성관리와 분산조종은 주문자싸이트에 설치된 판본이 (평가된) 주기계의 복사본과 같다는것을 확증하게 된다. A1클라스로 평가된 제품들은 극히 적다. 현재는 비록 이 클라스로 분류된 제품들이 존재함에도 불구하고 A1급 제품목록에는 2개의 망요소들만이 나타난다. 오렌지부크가 씌여 질 때는 체계방식과 검사, 형식적인 묘사와 검증에 대한 보다 많은 요구들과 믿음성 있는 설계환경을 가지는 A1급이상의 보다 높은 준위 담보클라스들까지도 정의하도록 고려되여 있었다. 그러나 보다 낮은급의 담보준위에 대해서조차 복잡한 쏘프트웨어제품을 평가하기 어렵다는것이 알려 지면서 이 방향에로 사업을 진척시키지 않고 있다.

제3절. 신용망해석

신용망해석(TNI)(Red Book)[107]은 오렌지부크에서 소개한 개념과 용어들로써 망보안을 설명하였다. 이 책은 오렌지부크의 일반화이며 이 일반화가 어디서 끝나는가 보는것이 흥미 있다. 실례로 레드부크는 망의 제한된 클라스에 국한하여 오렌지부크에서 제시되지 않은 문제들을 해명하며 ISO보안구성방식(ISO 7498-2[51])와 일정한 범위에서 경쟁한다. 따라서 레드부크를 오렌지부크와 그후에 제안된 [117]과 같은 새로운 기준사이의 련결을 지어 주는것으로 고찰할수 있다.

콤퓨터망들을 단순히 어떤 콤퓨터체계의 특정한 실례라고 가정하자. 그러면 오렌지부크의 방법들을 적용할수 있다. 레드부크는 이러한 가정을 검토하고 즉시에 다음과 같은 망의 두가지 류형을 구별하도록 한다.

- 여러가지 권한들과 방책들, 관리 등을 가지는 독립적인 요소들의 망(호상련결되고 인정된 자동화된 정보체계). 이러한 망에 보안을 실시하는것은 매우 어려운 일이다.
- 단일한 인정기관과 방책 그리고 망신용계산기지(NTCB)를 가지는 중심화된 망(단일한 신용체계).

레드부크에서는 두번째 형태의 망만을 취급한다. 콤퓨터망에서 보안기구는 여러 망요소들에 분산배치될수 있다. 때문에 다음과 같은 요인에 의해 새로운 보안문제들이 제기된다.

- 통신경로의 결함,
- 망요소들의 병렬적이며 비동기적인 조작.

이러한 문제들을 해결하는 암호화와 같은 기술적기구들은 오렌지부크에서 취급하지 않는다. 이러한 문제들을 취급하도록 레드부크를 두 부분으로 나누었다.

부분 I: 오렌지부크의 해석
부분 II: 기타 보안봉사들. ISO7498-2의 일정한 등급에 대응하며 현재는 신용망해석환경지도서[108]로 교체되였다.

1. 레드부크방책

망에서는 여러가지 실체들(사용자들, 봉사제공자들, 망조작자들, 체계관리자들, …)이 보안방책의 실행에 책임이 있다는것이 명백하다. 레드부크(Red Book)에서는 보안요구들을 진술하고 보안방책들을 결정하며 체계를 평가에 제기하는 실체를 주최자(sponsor)라고 부른다.

보안방책은 비밀엄수와 완정성을 취급한다. 보안방책은 접속지향적인 추상화의 도움으로 형식화되며 권한 있는 접속을 조종한다. 정점이름들은 C1클라스안에서 DAC그룹식별자로서 봉사할수 있다. MAC는 대화, 접속, 가상회로와 같은 새로운 개념적인 실체들에 적용한다. 식별과 인증은 《사용자가 떠나온》정점에 기초할수 있다. 검열궤적은 암호열쇠의 리용을 기록해야 한다. 대리사용자(감독적인 사용자)에 대한 론의는 대표문제의 한 실례이며 유일한 사용자에 관한 론의는 분포된 체계에서 이름짓기문제의 실례로 된다.

레드부크는 MAC가 본질적으로 중심화된 방책인 반면에 DAC는 체계내의 여러 정점들사이에 알맞춤히 분포시킬수 있다는것을 주장한다.

2. 완정성

제1장 1절 4는 용어완정성에 대한 단일한 정의가 없다는것을 보여 준다. 레드부크는 같은 문서안에 두개의 서로 다른 정의를 리용하여 이 점을 례증한다. 이 부분 1에서 완정성은 권한 없는 변경을 막기 위한 자료와 표식의 보호를 취급한다. 비바모형은 형식적인 완정성모형으로 제기되였다. 완정성은 또한 NTCB의 정확한 조작에 귀착된다. 부분 II에서 정보전송의 완정성은 통보문전송의 정확성 특히 통보문의 원천지와 목적지의 인증에 귀착된다. 암호기구는 자료의 고의적인 변경을 막는다.

3. 표식

망에서 위임접근조종은 위임완정성방책을 담고 있다. 완정성표식을 도입한다. 이러한 완정성표식은 실례로 객체가 정점들사이에서 전송된적이 있는가를 나타낼수 있다.

오렌지부크의 매우 협소한 해석은 보안표식이 주동체 또는 객체에 붙여 진 자료구조여야 한다는것을 요구한다. 이러한 명백한 보안표식들외에 다른것들도 있다. 객체를 보호하는데 암호문을 리용하면 암호열쇠가 암시적인 보안표식으로 될수 있다. 알맞는 열쇠에 접근한 주동체만이 객체에로 접근할수 있다. 따라서 열쇠에 대한 접근을 조종하는 표식은 암시적으로 암호화된 객체에 대한 접근을 조종한다.

4. 기타 보안봉사들

암호화와 통신규약은 보안봉사의 설계에 지원되는 근본요소들이다.

레드 부크의 고유한 보안봉사를 그림 9-1에 보여 주었다. 매 봉사에 대하여

- 기능
- 강도(strength)
- 담보

가 지적되여야 한다. 강도는 기구가 자기의 목적에 얼마나 잘 맞는가를 나타낸다.대표적 실례는 암호화알고리듬의 열쇠길이이다. 담보는 큰 범위의 검사를 하는 리론과 능숙한 쏘프트웨어공학실천, 확인과 검증으로부터 유도된다. 그 등급은 다음과 같다.

- 없음
- 최소(C1)
- 공정함(C2)
- 적당함(B2)
- 제공되지 않음-그대로

이 등급들은 오렌지부크의 등급들보다 더 응용의존적이며 안전성은 보다 약한것으로 보고 있다.

5. 평가클라스들과 구성규칙

오렌지부크 클라스들은 표준적인 보안요구들을 수용한다. 분할 C는 협동사용자들을 위한 환경을 지향한다. 여기서 담보에 대한 요구는 그리 높지 않다. 지향하는 환경에서 조작체계를 손상시키는것보다 보안을 우회하는것이 더 쉬운 방법이라는것을 추측할수 있다. 망에서 단일준위요소들로써 MLS 체계를 구축할수 있다. 그러나 C2 평가된 단일준위요소들은 분할 B체계에 요구된 담보준위로는 평가되지 않았을것이다. 따라서 B3클라스와 상응한 담보준위에서 위임접근조종이 아니라 검사를 위한 자유접근조종과 실시간경고를 제공하는 요소들을 위한 새로운 클라스 C2+가 도입되였다.

콤퓨터망은 마디나 통신규약 같은 명백히 식별할수 있는 요소들을 가지고 있다. 이 요소들의 보안등급으로부터 평가된 요소들로 구성된 망의 보안등급을 계산하는 평가방법을 찾을것이 요구된다. 실례로 맞춤한 강도의 식별통과규약을 통하여 통신하는 두개의

DAC요소들로 이루어 진 구성을 고찰하자. 이 구성의 등급을 두 요소의 등급들의 최소로 되게 정의할수 있다. 이것은 간단한 실례에서는 성립하지만 복잡한 망에서는 제한성을 가진다. 레드부크는 이 문제를 검토하고 평가가 보충적인 해석을 요구하는 구성들의 실례들을 제공한다.

망보안봉사	기준	평가
통신완정성		
인증	기능성	없음, 그대로
	강도	없음부터 적당함사이
	담보	없음부터 적당함사이
통신마당완정성	기능성	없음부터 적당함사이
	강도	없음부터 적당함사이
	담보	없음부터 적당함사이
거부 없음	기능성	없음, 그대로
	강도	없음부터 적당함사이
	담보	없음부터 적당함사이
봉사거절		
조작의 련속	기능성	없음부터 적당함사이
	강도	없음부터 적당함사이
	담보	없음부터 적당함사이
규약기초보호	기능성	없음부터 적당함사이
	강도	없음부터 적당함사이
	담보	없음부터 적당함사이
망관리	기능성	없음부터 적당함사이
	강도	없음부터 적당함사이
	담보	없음부터 적당함사이
보호손상		
자료기밀성	기능성	없음, 그대로
	강도	감수성수준
	담보	없음부터 적당함사이
전송량기밀성	기능성	없음, 그대로
	강도	감수성수준
	담보	없음부터 적당함사이

그림 9-1. 레드부크에서의 망보안봉사들

제4절. 정보기술보안평가기준

유럽정보기술보안평가기준[117]은 네데를란드, 영국, 프랑스, 도이췰란드가 국가적 보안평가기준을 정의하는 과정에 생겨 났다. 첫 초안이 1990년에 출판되였으며 정보기술보안기준(ITSEC)이 1995년 4월 7일에 유럽동맹평의회에 의하여 권고안으로 정식 인정되였다. 유럽문서로서 TSEC를 수많이 번역하였는데 유일하게 번역하는데서 난관들이 제기되였다.

ITSEC는 오렌지부크에 대한 각이한 해석들로부터 얻은 교훈에 기초한 론리적인 진보이다. 오렌지부크는 신축성이 없었으므로 ITSEC 는 새롭게 제기되는 보안요구들을 취급할수 있는 보안평가골격을 제공하려 하였다. 기능성과 담보사이의 련결이 파괴된다. 레드부크가 강도와 담보를 구분하였다면 ITSEC는 유효성과 정확성을 구분하였다. 유효성은 체계가 직면한 위험을 극복하는데 얼마나 적합한가를 나타낸다. 정확성은 체계의 발전과 조작에 관계되는 담보측면들을 포함한다.

1. 평가처리

보안체계는 물론 보안제품에도 기준이 작용한다. 제품은 기성으로 주어 질수 있다. 따라서 다만 그의 조작환경에 대한 일반적인 가정만을 할수 있다. 체계는 특정한 실세계환경에 전개된다. 두 항목을 다 포함하도록 하기 위해ITSEC는 평가목적(TOE)에 귀착된다.

평가의 주최자는 조작요구와 위험징후들을 결정하여야 한다. TOE의 앞으로의 보안목적은 법과 기타 규칙들에 의존한다. 그들은 필요한 보안기능성과 평가준위를 수립하였다. 보안목적은 평가와 관련되는 TOE의 모든 측면들을 서술한다. 그것은 TOE 의 보안기능성과 있을수 있는 위험징후, 달성할 목적, 리용할 보안기구들의 세부들을 서술한다. 보안목표는 다음의 항목들을 포함할수 있다.

- 보안목적 (조직적보안방책의 부분인 체계보안방책에서 서술됨),
- 체계환경에 대한 진술,
- TOE 환경에 대한 가정,
- 보안기능(기술적보안방책에서 서술됨),
- 보안기능들의 리론적기초,
- 요구되는 보안기구,
- 요구되는 평가준위(제9장 4절 4),
- 기구들의 최소강도의 주장된 등급.

매 평가준위에서 기준은 책임자가 평가자에게 넘겨 줄 항목들을 렬거한다.

평가자는 내용과 표현에 대한 어떤 요구가 만족되는가에 관심을 돌리면서 이 항목들이 제공된다는것과 그 항목들이 요구되는 증거를 명백히 제공 또는 보장한다는것을 담보한다.

주최자/개발자와 평가자사이의 긴밀한 협동처리를 권고한다. E1이상의 클라스들에 대해 평가자는 예비적으로 검사를 진행한 다음 개발자가 제공하는 결과들을 분석한다. 증명서는 TOE 가 그의 보안목표에 맞는가 어떤가를 진술한다.

2. 보안기능성

오렌지부크의 클라스정의에서 어떤 특정한 보호기구와 그의 기능 그리고 실현해야할 방책들의 리론적근거에 대한 몇가지 지적들을 볼수 있다. 그러나 이 세가지 문제들을 론하기 위한 조직적인 활동은 찾아 볼수 없다. ITSEC 가 바로 그것을 한다. 보안기능을 서술할 때 다음의 항목들을 진술해야 한다.

- **보안의 목적**: 왜 기능이 요구되는가?
- **보안기능**: 실제로 무엇을 하는가?
- **보안기구**: 그것을 어떻게 실현하는가?

보안기능들의 그룹화를 위한 일반적인 주제들은 다음과 같다.

- 식별과 인증,
- 접근조종,
- 책임추적가능성: 권한의 사용을 기록한다.
- 검사: 보안에 위험을 줄수 있는 사건들을 적발하고 조사한다.
- 객체의 재리용,
- 정밀성: 자료의 정확성과 일관성(이것은 완정성의 정의들중의 하나이다),
- 믿음성: 봉사의 일관성과 리용가능성,
- 자료교환: 국제규격 ISO 7498-2 에 귀착한다.

TOE의 기능은 개별적으로 또는 10개의 미리 정의된 기능클라스들중의 하나를 참조하여 지적할수 있다. 첫 5개 즉 F1-F5는 오렌지부크 클라스들의 기능성을 반영한다. 오렌지부크와 ITSEC사이의 사영을 그림 9-2에 제시하였다. 나머지클라스들은 특정한 응용령역을 가진다.

F6: 높은 준위 완정성 실례로 자료기지들
F7: 높은 준위 리용가능성
F8: 통신기간에 자료완정성
F9: 높은 준위 기밀성(암호화장치)
F10:높은 준위 기밀성과 완정성을 요구하는 망

이 클라스는 이이상 더 정의되지 않는다. ITSEC 는 새로운 기능클라스들을 추가하기 위하여 공개된 원리이다.

3. 유효성담보

보안기능들과 기구들의 유효성을 검사하자면 이것들이 지향하는 응용을 위해 《옳다》는것을 확증해야 한다. 완벽한 보안이 현실적인 목적은 아니다. 공개된 약점들에 관한 해결방도를 찾아야 한다. 공개된 약점들의 필연적인 결과가 주어 진 위험들을 허용할수 있다는것을 주장해야 한다. 유효성을 평가하자면 다음의 항목들을 검토해야 한다.

- 기능성의 적합성,
- 기능성의 결합(서로 다른 기능들의 량립성),
- 기구의 강도,
- 사용의 용이성,

- TOE의 구조안에서 보안약점의 평가: 실례로 보안시행기능을 우회하거나 손상시키는 방법들을 말한다.
- TOE의 조작내에서 보안약점을 평가.

4. 정확성의 담보

정확성의 담보는 7개의 평가준위 E0-E6을 리용하여 표현한다. E0-E6 평가준위들의 신뢰도는 증가하는 순서로 높아 진다. 이 준위들은 TOE의 구축과 조작에 귀착되며 개발자가 제공해야 할 문서목록과 평가자가 수행해야 할 행동절차의 목록을 서술한다. 이 문서들과 행동들은 다음의 항목들로 귀착된다.

- **개발공정**: 하향식방법, 보안요구들, 구조적설계, 세부적인 설계와 실현의 순서로 고찰한다.
- **개발환경**: 구성조종과 개발자보안을 포함한다.
- **조작**: 사용자와 관리자를 위한 조작문서와 전달, 구성, 기동, 조작을 포함하는 조작환경에 귀착된다.

7개의 평가클라스들의 주요특징들을 다음과 같이 요약할수 있다.

E0: 평가가 실패한 TOE들에 할당된 적당치 않은 담보
E1: TOE의 보안목적과 형식적인 서술. 검사는 TOE가 자기의 보안목적을 성취하였는가를 나타낸다.
E2: 세부적인 설계의 비형식적인 서술을 추가한다. 검사의 증거를 제공 받아야 한다. 구성조종과 조종식분배처리가 있다.
E3: 보안기능에 대응하는 세부적인 설계와 원천코드를 제공 받는다.
E4: 보안방책에 대한 형식적인 모형이 있어야 한다. 구조적이며 세부적인 설계를 위한 엄격한 방책과 표현이 요구된다. 약점분석은 이 엄격한 방책에 기초하여 진행해야 한다.
E5: 세부설계와 원천코드사이 밀접한 대응을 수립해야 한다. 약점분석은 원천코드를 리용한다.
E6: 보안방책의 형식모형과 일치하는 TOE 의 보안구조의 형식서술이 요구된다.

OB	ITSEC
D	E0
C1	F1+E2
C2	F2+E2
B1	F3+E3
B2	F4+E4
B3	F5+E5
A1	F5+E6

그림 9-2. 오렌지부크와 ITSEC사이의 대응

오늘 E3은 상업적인 보안제품들을 위한 가장 일반적인 평가준위이다. 안전한 조작체계나 자료기지관리체계는 표준적으로 F2+E3결합을 목표로 하고 있다. 자유접근조종이 충분한것으로 보이지만 C2클라스가 제공하는것보다 더 높은 준위 담보를 요구한다. 대부분의 평가는 여전히 오렌지부크클라스들과 류사한 기능클라스들에 관하여 수행된다.

제5절. 공통기준

유럽보안기준은 기능과 담보요구들을 분리하고 총적인 보안체계들을 고려함으로써 레드 부크와 신용자료기지해석(Trusted Database Interpretation)에 의해 로출된 문제들에 해답을 주었다.

ITSEC가 제공하는 신축성은 유리한 경우도 있지만 결함도 있다. 제1장 4절 3에서 강조한 콤퓨터보안의 근본난점을 상기해 보자. 보안전문가가 아닌 사용자들이 주어 진 보안목적이 자기들에게 적합한가를 어떻게 알수 있는가?

평가기준의 발전경로에서 다음순서는 련방기준(US Federal Criteria)[115] 이다. 이 책은 평가클라스들의 정의에서 보다 개선된 지도서를 제공하면서도 일부 신축성등급은 그대로 유지하는 다음단계의 평가기준이다. 련방규격은 제품의 평가와 평가클라스정의에서 기능과 담보사이의 련결을 주장하며 제품독립인 보호프로필을 통하여 오렌지부크의 신축성 없는 구조를 극복하려고 하였다. 보호프로필은 다음과 같은 5개의 부분들을 가진다.

- **서술요소**: 해결할 정보보호문제의 서술을 포함하는 보호프로필의 《이름》.
- **리론적근거**: 있을수 있는 위험, 사용환경, 사용에 대한 가정, 해결할 정보보호문제의 세부적서술, 보안방책에 대한 안내를 포함하는 보호프로필의 기본적인 정당화.
- **기능적요구**: 제품이 제공해야 할 보호경계를 설정. 이 경계내에서 예견한 위험들과 부닥칠수 있다.
- **개발담보요구**: 개발공정과 개발환경, 조작지원과 개발증거를 포함하는 초기설계로부터 실천단계까지의 모든 개발단계.
- **평가담보요구**: 평가의 형태와 강도를 서술한다.

보안평가가 사회적으로 인기를 끌게 하자면 평가증명서가 될수록 널리 인정되여야 한다. 이 방향에서 첫 단계는 평가기준의 공통모임을 합의하는것이다. 이로부터 국가적 보안평가를 책임진 여러 조직들이 공통기준편집위원회(Common Criteria Editing Board, CCEB)에서 서로 협력하여 CSEC, ITSEC, CTCPEC, 련방기준과 같은 현재 리용하고 있거나 또는 새로 만드는 평가기준들을 하나로 관통시키기 위해 노력한 결과 공통기준 [26] 을 만들어 냈다. 최종목적은 ISO규격의 형태로 국제적으로 인정되는 기준모임을 만드는것이다.

공통기준은 여러 선행한 기준들의 내용을 포함한다(결과 독자들은 방대한 문서들과 맞다들게 된다). 공통기준은 또한 ITSEC 의 총적인 신축성을 포기하고 보호프로필과 미리 정의된 보안클라스들을 리용하는 련방기준을 리용하고 있다. 이 클라스들의 정의는 보안목적, 리론적토대, 우려되는 환경, 앞으로의 응용상 주의점들에 대한 정보를 포함하고 있으므로 사용자들을 위한 지도서로 된다. 오렌지부크의 클라스들과 류사한 일부 클라스들이 실례로 주어 지지만 새로운 보안클라스들을 추가하는 공정이 있을것이라고 예상된다.

제6절. 품질규격

검사식평가의 최종단계는 제품 그자체에 대한 평가보다도 그 제품이 어떻게 개발되는가를 평가하는것이다. 그러면 회사는 《안전체계의 보증된 생산자》로 된다. 이러한 방책은 품질조종의 령역에서 널리 증명되였다. ISO 9000 과 같은 규격들은 자기들의 제품들을 보증하기 위하여 내부품질관리와 외부품질담보을 어떻게 할것인가를 권고한다. 일부 제작자들은 사용자들이 ISO 9000 품질표식을 특정한 제품의 보안증명서보다 훨씬 더 믿기때문에 보안평가는 이 방향으로 나가야 한다고 주장하고 있다.

이러한 제안이 안전체계개발자들의 인기를 끌고 있다. 평가비용이 훨씬 줄어 드는것이다. 만일 안전체계개발자들이 이 제안을 받아 들인다면 안전체계사용자들은 손해를 보게 되는가? 이것은 선행경험으로는 알수 없다. 결국 증명서는 체계가 파괴되지 않는다는 담보가 아니다.

그러므로 개별적으로 평가된 제품들이 인정된 개발자들이 만든 제품보다 더 완성된 보안을 제공하는가를 판단하기 위한 자체의 기준을 가지고 매개 평가기구들을 대해야 한다.

제7절. 평가비용

보안평가는 정부요구에 의하여 만들어 진 값 비싼 공정이라는 비난을 받았다. 이러한 의견들은 유럽콤퓨터제작자련합(ECMA)[7]에서 출판하는 보고서에서 제기되였다. 이 보고는 비용과 생산성, 보안사이의 균형에 대하여 주장하면서 이 요인들중 임의의 두개 요인들은 세번째 요인과 반작용한다는것을 강조하고 있다. ECMA는 현재 IT보안을 단순한 기술적문제로 취급하는것을 반대하며 보안체계관리의 중요성이 무시되는 조건에서 보안평가에 헛된 노력을 계속하여 생기는 투자의 불균형을 관찰한다.

현재 평가공정에 대한 그들의 비평에서 평가비용(개발비용의 10~40%)과 이 평가가 완성될 때까지의 시간지연이 리해관계의 령역이다. 언급된 기타문제들은 다음과 같다.

- 기준해석에서의 애매성과 기준의 점차적인 변형,
- 평가된 제품의 새로운 판본을 재평가하는 비용,
- 평가공정의 기밀성.

ECMA는 조작체계보안의 토대를 수립하는데서 오렌지부크의 C2클라스의 역할을 인정한다. C2는 주로 정부의 보안요구에 맞는것이므로 수많은 통과암호식의 고유한 요구들은 물론 식별과 인증, 접근조종, 책임추적가능성과 검사, 객체의 재리용, 정확성, 봉사의 믿음성을 고려하는 상업적인 기능성클라스(COFC)가 참고문헌[6]에 제안되였다. 보안평가에서 ECMA를 ISO 9000 과 같은 품질규격으로 볼것을 권고한다.

끝으로 증명서들은 어떤 제품의 특정한 판본판과 특정한 구성에 적용한다는것을 명심해야 한다. 실제의 설치에서는 대체로 서로 다른 판본과 서로 다른 구성이 리용되므로 엄밀히 말해서 증명서는 직접적인 보안담보를 할수 없다.

이 장의 문헌안내

보안평가와 보안모형의 형식화의 초기력사는 [93]에서 볼수 있다. 오렌지부크 클라스들의 견해와 보안평가에 대한 실례들은 참고문헌 [30]에 있다. Blacker 에 대한 간단한 설명은 [157] 에서 찾아 볼수 있다. 평가기준 , 부속문서들과 평가된 제품의 목록을 보여 주는 Web싸이트들은 다음과 같다.

http://ww.radium.ncsc.mil/teep/process/faq.html　TCSEC의 페지이다. 여기서도 평가제품들의 목록을 찾아 볼수 있다.

http://www.itsec.gov.uk　ITSEC의 페지이다.

ftp://ftp.cse-cst.gc.ca/pub/criteria/CTCEDC　카나다평가기준의 페지이다.

http://csrc.ncsl.nist.gov/cc　공통기준의 페지이다.

ITSEC보증제품들의 목록은 ［136］에 출판되였으며 6개월마다 갱신된다.

여러준위안전체계들의 담보와 잠복통로측면들에 대한 사례연구는 ［74］에 있다.

련습문제

1. 보안평가는 목표가 변화되는 경우에도 진행되여야 한다. 제품개발은 하나의 특정한 갱신판이 평가될 동안 한자리에 머물러 있지 않는다. 평가증명서들을 어떻게 최신으로 유지하겠는가?

2. 보안제품들은 목표가 변화되는 경우에도 들어 맞아야 한다. 위협환경이 제품의 수명기간에 변한다. 변화하는 위협환경에서 증명서들을 최신으로 유지하는 항비루스제품들의 평가기구를 어떻게 설치하겠는가? 기구에는 조작체계의 평가에 포함시켜야 할 요소들로서 어떤것들이 있는가?

3. 추가된 값을 제공하는 보안평가로부터 무엇을 예견하는가?

4. 평가기준은 보안을 모르는 사용자들이 고유한 보안요구를 만족시킬수 있게 도와 준다. 보호프로필이 이 문제를 위한 옳은 해결방도로 되는가?

5. ITSEC 는 보안체계의 평가를 포함한다. 상담(consultant)은 의뢰자들에게 그들의 보안문제를 해결하기 위한 조언을 준다. 상담을 청하는것이 좋은가 평가가 좋은가? 평가는 상담을 청하는것보다 어떤 우점을 가지는가?

6. 방화벽을 위한 보호프로필을 작성하시오.

7. 잠복통로의 블로크화와 감시를 위한 선택을 검사하시오. 잠복통로의 블로크화에 의하여 영향을 받는 체계의 리용능력은 어떠한가?

제3편. 분산체계

제10장. 분산체계의 보안

콤퓨터보안은 방책이 단일하고 체계행정이 단일하며 보안시행체계가 하나인 경우에는 《쉬워 진다》. 앞에서 고찰한것처럼 이러한 유리한 환경에서조차 보안을 그대로 실현하는것은 비현실적인것으로 되지만 그래도 분산체계보다는 훨씬 좋은 환경에 있다고 말할수 있다.

여기서는 어떤 망으로 결합된 콤퓨터의 모임을 간단히 분산체계(또는 이종체계 혹은 련합체계)라고 표현한다. 분산체계의 구성요소들은 서로 다른 조직들의 지배하에 놓여있게 되므로 보안방책도 서로 차이날수 있다. 리상적인 분산체계인 경우 사용자들은 자기들이 리용하고 있는 봉사와 대상이 놓여 있는 위치를 알려고도 하지 않는다.

목적

- 분산체계에서 제기되는 보안의 기초적인 문제들을 평가한다.
- 분산체계의 보안이 어떻게 발전해 왔는가를 본다.
- 분산체계에서 보안기구를 실현할 때 체계의 어느 층이 가장 적합한가를 검토한다.
- 분산체계에서 현재 도입된 보안기구들의 전망과 있을수 있는 착오를 미리 추측해 본다.

제1절. 소개

계산환경을 변화시킬 때마다 사용자는 설치된 보안기구의 적합성을 재평가하지 않으면 안된다. 집중체계로부터 분산체계에로 넘어 갈 때 보안에 반드시 영향이 미치게 되므로 분산체계보안을 검토할 때에는 먼저 집중체계에서 보안을 뒤받침해 주고 있는 모든 암시적가정들을 리해하는것이 본질적인 문제로 된다.

변화가 어떻게 사용자에게 영향을 주는가를 보기 위하여 통과암호에 의한 인증방법을 간단히 고찰하자. 통과암호들은 기계에로의 회선이 고정된 말단장치우에서 사용자가 작업하는 경우에는 실용적이라고 볼수 있다. 이때 말단장치와 기계사이의 련결이 보호되여 있으므로 통과암호를 도청할수 없고 통보문을 변화시키거나 삽입 또는 대화를 진행할수 없다고 확신할수 있는 타당한 근거가 있게 된다. 분산체계에서는 통신회선의 보호에 대한 이러한 가정을 준수하기 어렵다. 그럼에도 불구하고 통과암호들은 분산체계에서 아직까지도 가장 대중적인 인증기구로 되고 있다.

10년전까지만 하여도 분산체계에서 보안이라고 하면 주로 인증과 결부시켜 보는것이 기본이였지만 오늘날에는 내부망들의 경계를 보호하는 방화벽과 분산대상에 대한 접근을 보호하는 보안구성방식이 주류로 되고 있다. 이외에도 분산체계의 본성으로부터 사용자

권한의 포기와 부여,신용 그리고 보안과 분산체계의 다른 특징(실례로 믿음성 및 리용성) 사이의 호상작용문제가 보안의 문제로 되고 있다. 분산체계의 현 상태를 살펴 보기 위하여 먼저 분산체계에서의 일반적인 보안방책과 보안시행을 고찰한다.

1. 보안방책

분산체계에서 사용자들은 자기들이 접근하려는 대상인 마디에 반드시 등록되여야 한다는 법은 없다.

- 어떤 방법으로 사용자들을 인증하겠는가?
- 무엇에 기초하여 접근조종을 결정하겠는가?

다음의 3가지 선택항목들은 이 두가지 질문에 해답을 주고 있다. 사용자인증 특히 접근조종은 이 선택항목들에 기초하여 진행될수 있다.

- 사용자신원,
- 사용자가 조작하는 망주소,
- 사용자가 호출하고 있는 분산봉사(접근조작).

Unix는 첫번째 선택항목을 써서 **ftp** 또는 telnet와 같은 원격접근봉사를 진행한다. ftp프로그람은 Unix체계들사이에서 파일들을 전송하며 telnet는 원격가상말단을 창조한다. 이 두 프로그람은 사용자에게 사용자이름과 통과암호의 입력을 재촉한다. rlogin프로그람은 telnet와 류사하게 동작하는데 현재의 사용자이름을 자동적으로 전송하며 통과암호의 입력만을 재촉한다. 이 3가지 경우에 모두 표준적인 Unix접근조종기구를 적용할수 있지만 망에 의한 통신은 새로운 불리한 점들을 산생시키게 된다. 뒤에서 보다 적합한 인증방안을 론의한다.

만일 사용자신원에 근거하여 접근조종이 결정된다면 사용자접근권한을 어떤순서로 결정해 주겠는가? Unix환경에서는 사용자특권(뿌리) 그리고 원격마디로부터 오는 프로그람(SUID프로그람)의 특권과 그 소유권의 기능을 결정해 주어야 한다. 여러준위보안에서 표식화방식들은 분산체계의 마디들사이에서 차이가 있을수 있다. 오렌지부크와 레드부크에는 마디사이의 자료전송을 위한 방책이 제시되여 있으며 여러가지 립도로 된 보안기호들이 표기되여 있다.

악조건인 경우에도 분산체계에서 접근조작은 새로운 의미를 산생시킬수 있다. 다음과 같은 정황을 상상해 보자. 사용자가 원격봉사기에 보관되여 있는 자료에 대하여 읽기요구를 보낸다고 하자. 이때 봉사기는 사용자체계에 련결되여 있는 출력통로에로 자료를 쓸것이다. 봉사기는 읽기접근규칙 또는 쓰기접근규칙중에서 어느것을 적용하게 되는가?

체계에서 확정적인 마디들로부터 오는 사용자들은 다시 인증할 필요가 없다고 결정해도 된다. Unix에서는 .rhosts파일에 신용 받는 주마디(trusted hosts)로 표기할수 있다. 또한 신용 받는 사용자(trusted users)를 정의할수도 있는데 이 사용자는 통과암호를 제공하지 않아도 되며 rsh(remote shell)지령의 사용을 허용 받는다. 이러한 특징들은 매우 단순한 단번서명체계를 제공하지만 통보문인증의 질에 결정적으로 관계되며 제8장 7절 1에서 설명한바와 같이 신용 받는 령역에로 사용자를 허락하는 초기실체인증에도 관계된다. Windows NT에서 신용관계는 신용 받는 령역에 있는 사용자들이 신용하는 령역의 자원에 접근하는 보다 정교한 수단을 제공하고 있다(제7장 5절 4).

2. 위임

분산체계에서 통제된 호출(controlled invocation)은 문자그대로 새로운 차원을 의미한다. 사용자는 국부마디에 가입한 다음에 원격마디의 프로그람을 실행시킬수 있다. 원격마디의 자원에 접근하기 위하여 프로그람은 해당한 접근권한을 필요로 한다. 사용자접근권한이 프로그람에 부여되고 이 접근권한으로 프로그람이 원격마디우에서 동작하게 된다. 이 처리를 위임(delegation)이라고 한다.

이때 원격마디우의 프로그람은 사용자에 의하여 위임된 모든 접근권한을 가지고 동작하고 있다. 분산체계에서 사용자들은 자기들이 거의 조종할수 없는 마디에 모든 권한을 준데 대하여 안심할수 없다. 원격마디우에서 보호가 약하면 공격자는 사용자접근권한을 탈취하여 비법적인 목적에 리용할수 있다. 따라서 사용자들이 자기들이 위임하는 권한들을 조종할수 있고 책임추적가능성기구(accountability mechnism)들이 위임된 접근권한의 사용을 분간할수 있는 체계라면 보다 좋을것이다.

대중적인 봉사에서는 원격봉사요구를 처리하기 위하여 대리사용자(proxy user)들을 창조할수 있다. 먼저 원격사용자에게 봉사를 진행할수 있는 권한이 주어 졌는가를 검사한 다음 원격사용자가 위임한 권한대신에 그자신의 권한으로 동작하게 하면서 대리사용자가 요구된 동작을 수행하게 한다.

3. 보안시행

일단 방책들을 분류하였다면 이것을 어떻게 시행하겠는가를 결정해 주어야 한다. 명백히 다음과 같은 질문이 제기된다.

- 사용자를 어디서 인증하겠는가?
- 접근조종을 어디서 결정하겠는가?

여기서 《어디서》라는 질문은 두가지 방법으로 해석할수 있다. 제1장 5절에서 본 다섯번째 설계결심을 고려하면 보안을 집중적으로 실현하겠는가 또는 국부적으로 실현하겠는가를 결정할수 있다. 집중적인 경우에는 커베로스(Kerberos)(제10장 2절 1)와 같은 인증봉사기와 입장표수여봉사기를 의뢰하거나 방화벽을 설정하여 내부망에 대한 접근을 조종할수 있다(제13장 4절). 국부적인 경우에는 디지털사의 수자식분산체계보안구성방식(제10장 2절 2)에서와 같이 보안시행은 개별적인 마디우에 있는 조작체계에 맡긴다. 또한 다른 방법은 제1장 4절 2에서 본 두번째 설계결심에 근거하여 분산체계에서 보안을 실현하는 가장 적합한 층을 결정할수 있는데 이 방법은 제10장 4절에서 고찰한다.

제2절. 인증

망으로 전송된 보호되지 않은 통과암호들은 하나의 명백한 약점으로 된다. 이러한 약점의 리용은 쉽게 자동화될수 있다. 통과암호탐지자(password sniffer)는 망통신을 주의 깊게 살피다가 통과암호와 기타 보안에 관계되는 정보들이 들어 있는 파케트들을 추출하는 프로그람이다. 그러므로 분산체계보안을 실현하는 첫 단계에서는 보다 좋은 인증방안을 검토하게 된다. 아래에서는 집중 및 국부보안시행의 두가지 방안을 실례로 하여 고찰한다.

1. 커베로스

커베로스체계는 1980년대에 아테네(Athena)개발계획의 테두리에서 매써츄쎄쯔공과대학(MIT)에 의하여 개발되였다. 아테네는 MIT의 범위를 벗어 나 대학생들에게 콤퓨터자원을 제공하고 등록자리설정과 같은 추가적인 관리기능을 포함하였다. 커베로스가 제기한 모험과 위협은 [103]에서 서술되여 있는데 그 일부를 인용하면 다음과 같다.

> 환경은 실례로 은행거래, 기밀에 속하는 정부자료,대학생성적, 위험한 실험의 조종 등과 같은 민감자료나 모험이 큰 조작에는 적합하지 않다. 모험은 주로 권한이 부여되지 않은 사람들이 자원을 통제없이 리용하거나 체계자원 또는 사용자자원의 완정성위반, 개인용파일들을 되는대로 보는것과 같은 사적비밀의 란폭한 침해이다.

커베로스는 이때부터 사용자들의 인정을 받게 되였으며 특히 Internet RFC1510과 같은 분산체계의 인증에서 공업규격으로 채용되였다[80]. 커베로스는 분산체계에서 봉사를 위해 의뢰기들을 인증한다. 입장표(ticket)와 중앙보안봉사기(central security server)라는 개념을 받아 들여 인증체계를 구성하였다. 《당사자》(principal)라는 말은 망통신에 참가하는 의뢰기(사용자)들과 봉사기를 의미한다.

- **커베로스인증봉사기(KAS):** 가입할 때 당사자들을 인증하고 입장표들을 내보낸다. 이것은 일반적으로 하나의 체계가입대화에 유효하며 당사자들이 입장표수여봉사기로부터 다른 입장표들을 얻을수 있게 한다.인증봉사기를 때로는 열쇠분배쎈터(KDC)라고도 부른다.
- **입장표수여봉사기(TGS):** 인증을 요구하고 있는 당사자에게 망봉사에로의 접근을 주는 입장표를 내보낸다.

커베로스는 니드햄-슈뢰더(Needham-Schroeder)열쇠교환규약(제12장 3절 3)에 기원을 두고 있다고 볼수 있다. 암호화에는 DES와 같은 대칭암호화체계가 쓰이고 있다. 사용자들은 이름과 통과암호에 의하여 인증되지만 통과암호들은 망을 통하여 전송되지 않는다. Internet RFC 1510은 통신규약이 리행될수 없을 때 발생하는 오유통보문을 비롯하여 Kerberos Version 5 의 구체적인 명세를 제공하고 있다. 고찰을 쉽게 하기 위하여 커베로스체계의 통보문에서 일부 자료마당을 생략하고 통신규약이 성공적으로 완료된 경우만을 서술한다.편의상 기호들을 다음과 같이 나타낸다.

K_a:	사용자 A의 비밀암호화열쇠. 이것은 사용자 A의 통과암호로부터 한방향알고리듬에 의하여 유도된다. KAS는 K_a 의 복사본을 가진다.
K_{tgs}:	TGS와 KAS 에 의해 공유되는 비밀열쇠
K_b:	봉사기 B와 TGS가 공유하는 비밀열쇠
$K_{a,\,tgs}$	A 와 TGS사이에서 리용하기 위하여 KAS에 의하여 창조된 대화열쇠
$K_{a,b}$:	A와 B사이에서 리용하기 위하여 TGS에 의하여 창조되는 대화열쇠
$eK(X)$:	K열쇠밀에서 암호화된 자료파케트 X
$N1,N2$:	응답공격을 막기 위한 림시값(우연도전)
L_2,L_2:	입장표의 유효기간(수명)
T_1,T_2,T_3,T_4:	입장표 또는 인증자의 창조시간
$\text{Ticket}_{a,tgs}$:	A가 TGS에서 사용하기 위해 KAS에 의하여 창조된 입장표
$\text{Ticket}_{a,b}$:	A가 B에서 사용하기 위해 TGS에 의하여 창조된 입장표

그림 10-1은 의뢰기 A가 봉사기 B 에 접근하여 A와 B사이의 호상인증을 수립하기 위하여 설정해야 할 단계를 보여 준다.
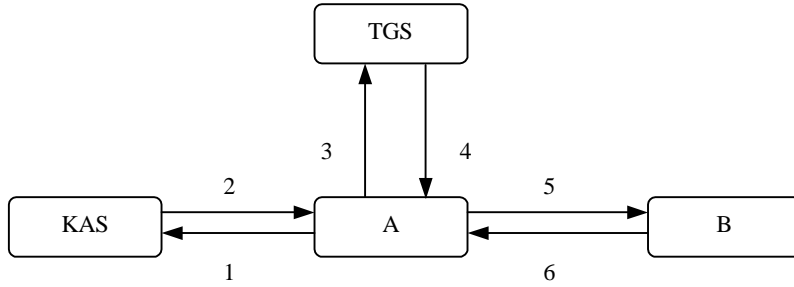
그림 10-1. 커베로스인증규약

통보문(1) $A \rightarrow KAS$ A, TGS, L_1, N_1

통보문(2) $KAS \rightarrow A$ $eK_a(\mathrm{TGS}, K_{a,tgs}, \mathrm{Ticket}_{a,tgs}, L_1, N_1)$

통보문(3) $A \rightarrow TGS$ $A, B, L_2, N_2, \mathrm{Ticket}_{a,tgs}, eK_{a,tgs}(A, T_3)$

통보문(4) $TGS \rightarrow A$ $eK_{a,tgs}(B, K_{a,b}, \mathrm{Ticket}_{a,b}, L_2, N_2)$

통보문(5) $A \rightarrow B$ $eK_{a,b}(A, T_4)$, $\mathrm{Ticket}_{a,b}$

통보문(6) B→A $eK_{a,b}(T_4)$

대화를 시작하기 위하여 사용자 A는 국부주마디에 가입하여 사용자이름과 통과암호를 입력하고 입장표수여봉사기의 봉사를 요구한다. 그다음 A의 신원, TGS의 이름, 요구된 입장표의 유효기간, 림시값들이 들어 있는 통보문(1)을 평문(cleartext)으로 KAS에 전송한다. KAS는 대화열쇠 $K_{a,tgs}$와 다음과 같은 입장표를 발생한다.

$$\mathrm{Ticket}_{a,tgs} = eK_{tgs}(K_{a,tgs}, A, T_1, L_1)$$

대화열쇠, 입장표, 림시값 N_1는 A의 비밀열쇠 K_a 에 의하여 암호화되여 통보문(2)로 A에 돌려 진다. A의 주마디에서 통과암호로부터 K_a를 다시 구성하고 대화열쇠 $K_{a,tgs}$를 얻는다. 이때 의뢰기 A는 인증자 $K_{a,tgs}(A, T_3)$을 창조하여 인증자, 입장표, 요청된 유효기간 L_2, 림시값 N_2, 봉사기의 이름을 통보문(3)으로 TGS에 보낸다.

TGS는 K_{tgs}를 리용하여 입장표를 복호하고 국부시계에서 입장표가 유효한가를 확인한다. 그다음 TGS는 표로부터 오는 $K_{a,tgs}$열쇠를 리용하여 인증자를 검사한다. 모든 확인이 잘되여 가면 TGS는 대화열쇠 $K_{a,b}$와 다음의 입장표를 생성한다.

$$\mathrm{Ticket}_{a,b} = eK_b(Ka, b, A_1, T_2, L_2)$$

통보문(4)에서 대화열쇠 $K_{a,b}$와 $\mathrm{Ticket}_{a,b}$는 A에 보내지며 이 대화열쇠가 주어 진 조건에서 암호화된 A는 TGS와 공유한다. 의뢰기 A는 이 암호화된 입장표를 기억하고 새로운 대화열쇠 $K_{a,b}$를 복호한다. 통보문(5)에서 A는 B에 대화가 인증되였는가를 물어 본다. 이 통보문은 $\mathrm{Ticket}_{a,b}$와 대화열쇠 $K_{a,b}$에 의하여 구성된 새로운 인증자를 포함하고 있다. B는 입장표를 복호하고 그의 유효성을 검사한후에 대화열쇠 $K_{a,b}$를 얻는다. 그다음 B는 $K_{a,b}$로 인증자를 복호한다. 복호가 성공하고 시간이 확인되면 B는 곧 통보문(6)

에서 대화열쇠에 의하여 수신되고 암호화된 마지막시간도장으로 응답한다. A는 시간도장을 복호하고 그것을 자기의 T_4 복사본과 비교한다. 만일 일치하면 B는 인증되였다.

취소

당사자로부터 어떻게 하면 접근권한을 취소할수 있는가? KAS와 TGS의 체계관리자는 접근권한이 더는 당사자들에 쓰이지 않도록 하기 위하여 자기의 자료기지들을 갱신하지 않으면 안된다. 이와 같이 접근권한은 당사자가 체계에 가입하거나 TGS로부터 입장표를 요구하는 다음번의 대화를 위하여 취소되였다. 그러나 당사자가 이미 점유한 입장표들은 만기가 될 때까지는 유효하다. 례하면 KAS의 입장표들의 수명은 대체로 하루인데 이것은 다른 한가지 TOCTTOU문제의 실례로 된다.

편리성과 보안의 관계는 이률배반관계에 놓이게 된다. 만일 TGS가 유효기간이 긴 입장표를 내보낸다면 당사자는 TGS에 자주 접근할 필요가 없고 TGS는 사용자들에게 지나치게 큰 영향을 주지 않는 조건에서 경우에 따라 비직결상태로 될수 있다. 그러나 접근권한을 포기하면 지연이 더 길어 지게 된다. 만일 TGS가 수명이 짧은 입장표를 내보낸다면 당사자들은 보다 일정한 기간마다 자기의 입장표를 갱신하여야 하므로 보안봉사기의 리용성은 체계의 성능을 위하여 보다 중요한 문제로 된다.

지역

커베로스인증봉사기는 커베로스지역의 중심부에 위치한다. 커베로스지역이란 봉사기집합에로의 접근을 조종하는 단일한 행정령역을 말한다. 커베로스가 기동하자면 당사자들은 KAS에 의하여 등록되여야 하며 TGS는 접근조종정보를 수신하여야 하며 모든 필요한 열쇠들이 보안관리자에 의하여 제자리에 놓여 있어야 한다. 커베로스는 집중보안체계가 가지는 우점을 다 가지고 있다. 제한된 보안봉사기에 의하여 유일한 보안방책이 리행되게 된다. 체계설정이 보안방책에 따른다는것을 검사한 다음 필요하다면 변경들을 실현하는것은 비교적 쉽다.

커베로스지역들사이의 접근이 가능하게 되면 지역들의 계층을 창조할수 있다. 이때 서로 다른 대화열쇠들이 인증봉사기의 《련결사슬》을 따라 쌍으로 교체되므로 인증수속과정에 품이 증가하게 된다.

요약

커베로스를 완전히 평가하기 위해서는 문제의 범위를 벗어 나 인증규약과 구체적인 암호화알고리듬의 강도에 대한 분석을 하여야 한다. 또한 그의 비암호화적보안특징을 검토하여야 한다. 아래에 이러한 연구에서 제기되는 문제들중 일부를 주었다.

- 통보문의 적시성(timeliness)을 시간도장(timestamp)으로 검사한다. 따라서 체계전반에 합리적인 동기화박자가 필요하며 박자 그자체는 공격으로부터 보호되여야 한다. 박자동기화를 보호하는 그자체가 인증을 요구할수 있다.
- 시간도장을 검사할 때 일부 박자가 일치되지 않아도 된다. 대표적인 5분간의 접수창은 길다고 볼수 있으므로 응답공격에 의하여 쉽게 리용 당할수 있다.
- 봉사기들은 직결식이여야 한다. KAS는 가입시 직결식으로 되여야 하지만 TGS는 입장표를 요구하는 시각에 필요하게 된다. TGS의 리용성에 대한 요구는 우에서 본바와 같이 완화시킬수 있다.

- 대화열쇠(대칭암호화장치에서)들은 커베로스봉사기(인증 및 입장표수여봉사기)에 의하여 발생된다. 대화열쇠들이 당사자들의 통신에서 리용되기때문에 봉사기의 믿음성에는 도청능력이 부정적으로 사용되지 않는다는 신용을 포함시켜야 한다.
- 커베로스는 특권의 위임(입장표)문제를 취급하지 않는다.
- 통과암호의 추측과 통과암호의 기만공격이 가능하다.
- 열쇠들과 입장표들은 의뢰기의 기계에 보관된다. 그러므로 커베로스보안을 위한 마디에 있는 보호기구만 믿을수 있다. 커베로스사용자가 단순한 말단에서 작업하는 경우에는 그리 큰 문제로 되지 않지만 PC나 다중사용자워크스테이션에서 커베로스가 동작하는 경우에는 사정이 달라 진다.
- 규약의 보안 그자체를 실현의 보안과 구별해 보는것이 중요하다. 실례로 Kerberos Version4에서는 열쇠의 발생에 약한 우연수발생기를 리용하였으므로 완전탐색방법을 적용하면 열쇠들을 쉽게 알아 낼수 있다.

2. DSSA/SPX

SPX는 디지털이큐이프먼트사(Digital Equipment Corporation)에서 개발한 분산체계보안구성방식(DSSA)의 한부분이다. DSSA는 워크스테이션으로 이루어 진 망을 위한 보안구성방식으로서 인증과 다른 보호수단들이 동시에 포함되여 있다[116,151,160]. 매개 마디는 자기의 보안방책을 리행한다. 사용자들은 자기들이 가입한 매개 마디우에 있는 조작체계를 믿어야 한다. 따라서 사용자들이 지금 사용하고 있는 마디들에 특권을 위임한다는 주장이 정당하다고 볼수 있다. DSSA/SPX는 Internet RFC 1507으로서 채택된 분산인증보안봉사방식으로 되였다.

SPX에서의 인증체계는 당사자의 이름과 장기비공개열쇠가 들어 있는 신임장과 당사자의 이름을 공개열쇠와 결합하는 보증서 그리고 인증통표로 이루어 져 있다. 신임장(credential)은 SPX안에 있는 사용자를 대표한다. 신임장은 봉사기에 기억될수 있으며 사용자통과암호가 초기화될것을 요구할수 있다. 또한 신임장은 사용자가 가지고 다니는 지능카드에 보관될수도 있다. 대화과정에 사용자의 초기등록과 인증은 서로 다른 봉사기에 의하여 진행된다.

- **보증권(CA):** 공개열쇠보증서를 내보내며 비직결로 처리할수 있다. 유효한 보증서를 내기 위하여서는 CA의 믿음을 받아야 한다.
- **보증서배포쎈터(CDC):** CA에서 나오는 보증서들을 보관한다. 이름지정봉사에 의하여 CDC의 기능을 제공 받을수 있다. CDC는 인증기간에는 직결처리되여야 한다.

CDC는 인증규약의 보안을 약화시킬수 없다. 무효인 보증서를 내는 경우에는 규약은 오유로 끝나게 될것이며 이런 측면에서 CDC를 믿어서는 안되지만 인증봉사의 리용가능성에 있어서는 CDC가 결정적이다. 실제적으로 실현할 때 CDC를 분배 받을수 있다. 보증서취소목록을 보관하거나 CDC자료기지에서 오는 보증서를 삭제함으로써 보증서를 취소할수 있는데 이 경우 CDC는 틀림없이 보증서를 정확히 삭제한다. 만일 보증서를 직결처리할것을 CDC가 요구하지 않는다면 보증서는 만기가 될 때까지 유효하게 되므로 개별적인 마디에서 접근권한을 취소할수 있다. SPX 의 약속기호는 다음과 같다.

- S_p : 당사자 P의 비공개서명열쇠,
- P_a, S_a: A의 장기공개열쇠와 장기비공개열쇠,
- P_a', S_a' : A의 단기공개열쇠, 단기비공개열쇠, A가 가입될 때 창조된다.

- $K_{a,b}$: A와 B사이의 사용을 위하여 대칭암호화알고리듬을 써서 A에 의하여 창조된 대화열쇠,
- $eK(X)$: 열쇠 K가 주어 진 조건에서 암호화된 자료파케트 X,
- $sK(X)$: 열쇠 K에 의하여 발생된 자료파케트 X의 수자식서명. 표기를 간단히 하기 위하여 서명을 보낼 때마다 이미 서명한 자료가 통보문에 있다고 가정한다. 그렇기 때문에 $sK(X)$는 X와 X우에서의 서명을 의미한다.
- T: 시간도장,
- L_c, L_t: 보증서나 입장표의 유효기간.

여기서 공개열쇠들은 암호화와 서명확인에 쓰이지만 비공개열쇠들은 복호화와 서명에 쓰인다. 그러나 실제적인 체계에서 표기는 복잡하다. 단기공개열쇠/비공개열쇠쌍은 장기열쇠쌍의 지나친 로출을 피하기 위하여 쓰이고 있다. 새로운 신임장을 창조하고 사용자신원을 단기비공개열쇠에 결합하면 위임이 실현될수 있다. 단기비공개열쇠는 사용자가 호출하고 있는 봉사에 적용될수 있도록 되여야 한다.

당사자 A와 B사이의 호상인증과정을 그림 10-2에서 보여 주었다. CA_a가 내보내는 당사자 B의 보증서는 A에 의하여 신용 받는 보증권으로서 다음과 같은 형태로 표시된다.
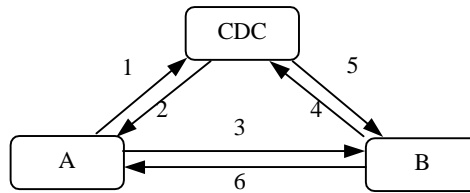
그림 10-2. DSSA/SPX 인증규약

보증서(B, CA_a)=$sS_{CAa}(CA_a,\ B,\ L_c,\ P_b)$

이때 통보문들은 다음과 같은 순서로 교환된다[151].

통보문(1) $A \rightarrow CDC$: B

통보문(2) $CDC \rightarrow A$: 보증서(B, CA_a)

통보문(3) $A \rightarrow B$: $A,\ eK_{a,\ b}(T,\ A),\ sS_a(L_t,\ A,\ P_a'),\ eP_b(K_{a,\ b}),\ eK_{a,\ b}(S_a')$

통보문(4) $B \rightarrow CDC$: A

통보문(5) $CDC \rightarrow B$: 보증서(A, CA_b)

통보문(6) $B \rightarrow A$: $eK_{a,\ b}(T)$

통보문(1)에서 요청자 A는 CDC에 검증자B의 장기공개열쇠를 요구한다. 그러면 CDC는 통보문(2)에서 보증서가 B의 신원을 공개열쇠 P_b와 결합시키고 있다고 응답한다. 공개열쇠 P_b는 CA_a의 공개검증열쇠를 리용하고 있는 A에 의하여 검증된다. 지금 A는 대화열쇠 $K_{a,b}$를 발생한다. 이 대화열쇠는 인증자 $eK_{a,b}(T, A)$, A의 단기공개열쇠 P_a'에 대한 서명입장표 $sS_a(L_t, A, P_a')$, 위임자 $eP_b(K_{a,b})$과 $eK_{a,b}(S_a')$를 창조하는데 쓰인다. 만일 위임권이 의도되지 않았다면 그대신에 위임자 $sS_a'(eP_b(K_{a,b}))$를 리용하여 열쇠 S_a'가 B에 로출되지 않도록 한다. 이 자료마당들은 통보문(3)에서 검증자에 보내진다.

통보문(4)와 통보문(5)에서 검증자는 CDC로부터 오는 요청자의 장기공개열쇠에 대한 보증서를 검색한다. 이 보증서에 의해 B는 요청자의 장기공개열쇠 P_a를 인증한다. B는 복호를 위하여 자기의 장기비공개열쇠를 리용하여 위임자중에서 대화열쇠 $K_{a,b}$를 검색한다. 그다음 B는 대화열쇠를 리용하여 인증자를 복호하고 시간도장을 자기의 국부동기박자와 비교한다. 다음에 B는 보증된 검증열쇠 P_a를 리용하여 P_a'을 포함하고 있는 입장표 $sS_a(L_t, A, P_a')$에 대한 서명을 검사한다.

최종적인 검사는 위임자의 형식에 관계된다. 위임자 $eP_b(K_{a,b}), eK_{a,b}(S_a')$가 주어진 조건에서 검증자는 S_a'을 검색하고 S_a 와 P_a'이 적합한 열쇠쌍이라고 확신한다. 만일 위임자 $sS_a'(eP_b(K_{a,b}))$의 전송이 완료되였다면 검증자는 P_a'에 의하여 $eP_b(K_{a,b})$에 대한 서명을 검사한다. 검사가 전부 성공하였다면 B는 인증자 $eK_{a,b}$ 로 응답하면서 걸음(6)에서 호상인증을 끝낸다. A는 대화열쇠를 리용하여 인증자를 복호화하고 자기한테 복사되였는가를 확인한다.

3. 개인용암호화주변장치

커베로스체계에서 사용자들은 자기의 통과암호를 머리속에 기억하여야 하며 응용봉사기들은 입장표수여봉사기와 공유하는 열쇠만을 필요로 하며 보호에 관계되는 기타 모든 정보들은 불과 몇개의 중심봉사기에 보관되여 있다.그러나 DSSA인 경우에는 사용자들은 비공개열쇠를 보유하여야 하며 보호에 관계되는 기타 대부분의 정보들은 국부적인 봉사기에 의하여 관리된다. 이때 중심봉사기들은 보증서들을 보관하는 보관소로만 된다.

한걸음 더 나아가 보호에 관계되는 그밖의 정보를 분산시켜 사용자가 기억시키게 할수 있다. 사용자들은 지능카드나 PCMCIA와 같은 개인용암호주변장치들(PCS)을 휴대하고 다니면서 이것을 리용하여 임의의 마디로부터 분산체계에 접근할수도 있다. PCP에는 사용자와 관련된 암호열쇠,접근조종파라메터, 암호화알고리듬의 실현방법들이 들어있을수 있다.

포테자(Fortezza)카드는 미국방성에서의 응용을 목적으로 개발한 PCMCIA카드로서 이 방법을 설명하고 있다. 사용자가 카드를 내보이면 싸이트보안원은 이 카드를 사용자방식으로 절환하기전에 사용자의 비공개열쇠를 넣는다. 이 카드는 PIN기구를 통하여 능동으로 된다. 포테자는 개별적인 사용자들이 자기들의 PCMCIA카드에서 보안에 관계되는 조작을 함으로써 어떤 마디에서나 안전하게 분산체계에 접근할수 있도록 하고 있다.

제3절. 보안 API

분산체계보안을 위한 포괄적인 구성방식들로 넘어 가기 위해서는 다음의 3가지 문제점들을 취급하여야 한다.

1. 분산체계에서 보안의 요구는 보통 순수한 인증을 초월하게 된다.
2. 분산체계의 서로 다른 요소들이 반드시 동일한 보안기구를 리용하여야 한다는 법은 없다(이 장에서 이미 사용자인증에서 쓰이는 두가지 서로 다른 방식들을 고찰하였다). 이때 보안은 다양한 보안기구우에서 실현되여야 한다.
3. 사용자들과 응용프로그람작성자들은 반드시 보안전문가가 아니라도 된다. 비전문가들에게 보안봉사를 제공하는 방법을 찾아 내야 한다.

쏘프트웨어공학은 이 세가지 문제를 론의하기 위한 충분히 정립된 전략을 제기하고 있다. 체계는 층으로 분해된다. 응용프로그람대면부(API)는 어떤 층에 있는 어떤 응용프로그람이 층아래에 있는 봉사를 호출할수 있게 한다. 구체적인 실현부분을 은페시킴으로써 API는 실현방법이나 지어는 암호화알고리듬의 선정과 같은 보안특정의 파제로부터 응용프로그람작성자를 해방하고 있다. 요구되는 보안에 대한 전문지식소유정도(보안지식)는 API보안을 비교하기 위한 합리적인 기준점으로 된다.

1. GSS-API

일반보안봉사API(Generic Security Service API, GSS-API)는 원래 커베로스나 분산인증보안봉사(Distributed Authentication Security Service, DASS)에 기초하여 분산보안구성방식들사이의 이식성을 보장하기 위하여 만들어 졌다. GSS-API는 련결지향응용을 위한 보안봉사와 결합하기 쉬운 대면부를 제공한다. 이러한 봉사들은 어떤 범위의 기구들과 규약에 의하여 실현될수 있으므로 여러 환경에 적용할수 있도록 원천준위에서 이식성이 있어야 한다. GSS-API설계에서 기본목적은 다음과 같다.

- **기구독립성**: GSS-API는 개별적인 장치에 독립인 강한 인증 및 기타 보안봉사와의 대면부를 정의한다. 보안봉사들은 대칭열쇠암호화(실례로 커베로스) 또는 공개열쇠암호화(X509)로 실현될수 있다.
- **규약환경독립성**: GSS-API는 리용된 통신규약에 관계 없으므로 넓은 범위의 규약환경에서 리용할수 있다.
- **설치장소범위의 적합성**: GSS-API의뢰기는 그것이 동작하는 체계우에서 결정되는 반드시 신용계산기지(TCB)둘레안에 있어야 한다는 제한은 없다.

GSS-API에는 두가지 론리적부분 즉 보안봉사모임과의 일반적대면(고유한 API)과 보안봉사를 보장하는 기구의 모임이 있다.

GSS-API 특성과 개념

전형적인 GSS-API호출자는 인증, 완정성 그리고(혹은) 기밀보안봉사와 자기의 통신방법의 보호 또는 기밀보안봉사통신을 보호하기 위하여 GSS-API봉사를 요구할 때 그자체가 통신규약으로 된다. 대면부는 국부체계에 상주하면서 서고를 통하여 GSS-API의 접근을 보장한다. 대면부는 매개의 기구와 결합하면서 응용으로부터 보안기구의 세부를 숨기고 자료변환과 호출의 역할을 한다. GSS-API 의 기본적인 보안요소들은 신임장, 통표, 보안문맥, 상태코드들이다. 대등망의 실체를 인증하기 위하여 대등망실체들사이의 보안문맥을 초기화하는 조작들은 통보문당자료원본인증(per-message data original authentication)과 자료완정성보호를 제공하는 조작들로 분리된다.

보안봉사를 제공하는데 쓰이는 기구들은 매개 대화의 시작에서 선택되며 대화기간에는 고정되여 있게 된다. GSS-API에 적용할수 있는 기구들은 선택에 영향을 준다. 여러개의 기구들이 쓰이는 곳에서는 호출자가 선택목록을 가리킬수도 있다. 그렇지 않으면 기정기구가 선택된다.

신임장

신임장(credentia)에는 대등망실체가 서로 보안문맥을 설정할 때 요구되는 보안관련 자료들이 있다. 신임장의 구조는 아직 규격화된것이 없다. 신임장의 내용을 정의하는것은 하위층에 남겨 두지만 서로 다른 기구가 동일한 구조를 사용할수도 있다. 신임장은 응용

의 보안을 유지하기 위하여 기초에 놓이는 체계에 따라 적당히 조절되도록 되여 있어야 한다. 신임장들은 GSS-API의 호출변수가 아니라 신임장조종프로그람의 참조에 의하여 넘겨 진다.

통표

GSS-API호출자에 의하여 그의 대면호출에 공급되는 자료들은 기구특정의 형식화된 통표(token)들로 변환되게 된다. GSS-API호출자는 국부GSS-API의 실현에 의하여 제공되는 이 통표들을 접수하고 이것을 원격체계우에 있는 대등망에로 전송한다. 이때 대등망은 처리를 위하여 수신된 통표를 자기의 국부GSS-API실현에 넘겨 준다. 통표들은 일반적으로 호출자에는 아무런 의미도 없으며 그의 특정형식은 리용되고 있는 개별적기구들에 상당히 의존한다.

보안문맥

보안문맥(Security Contexts)은 보안봉사의 관리에 관계되는 정보를 반영하고 있다. 보안문맥은 신임장을 리용하는 대등망들사이에 설정된다. 신임장의 동일한 모임 또는 다른 모임을 사용할 때 한쌍의 대등망사이에는 여러개의 보안문맥이 동시에 존재할수도 있다. 여러가지 신임장들을 리용하는 여러개의 보안문맥이 공존하면 신임장의 유효기간이 끝날 때 연장될수 있다.

상태코드

상태기발들은 어떠한 특징들이 필요한가를 가리킨다. 즉 mutual_reg_flag는 호상인증이 요구된다는것을 나타낸다. 보안문맥의 초기화가 완전히 끝나지 않았다는것을 나타내는 상태코드 실례로 GSS_ONTINUE_NEEDED에 의하여 보안문맥의 설정을 지원할수 있다. 대부분의 상태코드들은 기구에 무관계하지만 일부 코드들은 기초적인 보안기구에 따라 달라 질수 있다. 이 상태정보는 보안에 관계되는 호출에 대하여 제한된 범위에서 인과적인 검사를 진행할수 있는 가능성을 GSS-API호출자에게 주고 있다. 호출자의 과제는 호출이 정확한 보안문맥에서 발생되는가를 확인하는것이다. 관리과제와의 호출자련관의 확장을 보여 주기 위하여 RFC 2078에 서술된 모든 기본상태되돌림값들의 목록은 다음과 같다.

GSS_S_BAD_BINDINGS	통로결합이 정합되지 않았다.
GSS_S_BAD_NECH	요구한 기구가 지원되지 않았다.
GSS_S_BAD_NAME	제공된 이름이 맞지 않는다.
GSS_S_BAD_NAMETYPE	지원되지 않는 형의 이름이 제공되였다.
GSS_S_BAD_STATUS	입력상태선택기가 무효이다.
GSS_S_BAD_SIG	통표의 완정성검사가 무효이다.
GSS_S_CONTEXT_EXPIRED	지적된 보안문맥의 유효기간이 끝났다.
GSS_S_CREDENTIALS_EXPIRED	유효기간이 지난 신임장이 검출되였다.
GSS_S_DEFECTIVE_CREDENTIAL	결함 있는 신임장이 검출되였다.
GSS_S_DEFECTIVE_TOKEN	결함 있는 통표가 검출되였다.
GSS_S_FAILURE	GSS-API준위에서 서술되지 않은 고장.
GSS_S_NO_CONTEXT	지적된 보안문맥이 무효이다.
GSS_S_NO_CRED	제공된 신임장이 무효이다.
GSS_S_BAD_QOP	QOP값이 지원되지 않았다.

GSS_S_UNAUTHORIZED	조작에 권한이 주어 지지 않았다.
GSS_S_UNAVAILABLE	조작이 해당되지 않는다.
GSS_S_DUPLICATE_ELEMENT	2중화된 신임장요소를 요구하였다.
GSS_S_NAME_NOT_MN	이름에 다중기구요소가 있다.
GSS_S_COMPLETE	정상으로 완료되였다.
GSS_S_CONTINUE_NEEDED	루틴호출을 계속 요구한다.
GSS_S_DUPLICATE_TOKEN	2중화된 통보문당 통표가 검출되였다.
GSS_S_OLU_TOKEN	시간초과된 통보문당 통표가 검출되였다.
GSS_S_UNSEQ_TOKEN	재순서화된 통보문당 통표가 검출되였다.
GSS_S_GAP_TOKEN	건너뛰기한 선행통표가 검출되였다.

대면부서술

호출에는 4가지 형태가 있다.

- 신임장관리호출: 이것은 당사자가 신임장을 획득하였다가 해방하고 다양한 신임장정보를 질문할수 있게 한다.
- 문맥준위호출(context-level call): 이것은 문맥의 초기화,접수,삭제를 위하여 또한 문맥의 유효시간을 결정하고 문맥의 통표를 처리하기 위하여 쓰인다.
- 단위통보문(per-message)호출: 이것은 암호의 완정성과 기밀보호를 제공한다.
- 지원호출: 이것은 실례로 할당된 기억기를 해방하고 이름을 비교하는것과 같은 일반적인 정리작업(housekeeping) 및 지원루틴을 위하여 쓰인다.

RFC-2078 에서 정의된 GSS-API호출은 모두 다음과 같다.

신임장관리호출 :

GSS_Acquire_cred	사용을 위하여 신임장을 획득한다.
GSS_Release_cred	사용후에 신임장을 해방한다.
GSS_Inquire_cred	신임장에 대한 정보를 현시한다.
GSS_Add_cred	신임장을 증가하는 순서로 구성한다.
GSS_Inquire_cred_by_mech	단위기구신임장정보를 현시한다.

문맥준위호출 :

GSS_Init_sec_context	범위를 벗어 나는 보안문맥을 시동한다.
GSS_Accept_sec_context	범위안에 있는 보안문맥을 접수한다.
GSS_Delete_sec_context	더는 필요되지 않을 때에 문맥을 닫는다(flush).
GSS_Process_context_token	문맥에 관한 수신된 조종통표를 처리한다.
GSS_Context_time	문맥에 남아 있는 유효시간을 나타낸다.
GSS_Inquire_context	문맥의 정보를 현시한다.
GSS_Wrap_size_limit	GSS_Wrap통표크기의 한계를 결정한다.
GSS_Export_sec_context	문맥을 다른 처리로 전송한다.
GSS_Import_sec_context	전송된 문맥을 받아 들인다.

단위통보문호출:

GSS_GetMIC	완정성검사를 적용하여 통보문에서 분리된 통표로 수신한다.
GSS_VerifyMIC	통보문과 함께 완정성검사통표를 유효하게 한다.

GSS_wrap	서명을 선택권에 따라 암호화하고 밀봉한다.
GSS_Unwrap	밀봉을 해제하고 필요하다면 복호하고 완정성검사를 유효하게 한다.

지원호출:

GSS_Display_status	상태코드들을 인쇄할수 있는 형태로 변환한다.
GSS_Indicate_mechs	국부체계우에서 지원된 mech_type를 나타낸다.

2. API와 보안

보안봉사가 효과적으로 되자면 해당한 보안기구를 합리적으로 실현할뿐아니라 보안관리도 합리적으로 하여야 한다. 보안봉사가 암호화를 리용할 때에 열쇠관리는 보안관리의 중요한 부분으로 된다. 그러나 열쇠관리는 보통 봉사요구를 처리하고 있는 기간에 실행되는 과제가 아니라 봉사요구가 실행되기전에 처리되는 과제이다. 이것은 그림 10-3에서 보여 준 IBM보안구성방식에 반영되며 보안관리가 모형의 다른 층과 직교되게 놓여있는 별개의 문제와 같이 보인다.

물론 보안관리가 이 모형에서 모든 층에 관계된다는것은 사실이다. 설계자들이 열쇠관리의 이 측면을 무시하는 경우 문제가 생기지만 보안에 대한 지식이 없는 응용프로그람작성자들이 보안에 신경을 쓰지 않기를 기대할 때에는 계층적인 설계전략에 따르게 된다. 이를 위하여 열쇠관리기능을 가르고 응용대상의 책임범위내에서 부분별 과제를 명백히 나타내야 한다. 열쇠관리에서 고려할 인자들은 많으며 적어도 다음의 4가지를 반드시 고려하여야 한다.

- 열쇠발생
- 열쇠기억기
- 열쇠전송
- 열쇠사용
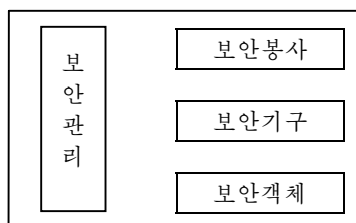
그림 10-3. IBM보안구성방식

마지막항목은 열쇠가 한가지 목적에만 사용되여야 한다는것을 의미한다. 실례로 수자식서명을 발생하거나 이것을 확증하는데 쓰일수 있지만 절대로 두가지를 동시에 쓸수 없다. 이러한 응용들은 재정분야에 널리 도입되였으며 실례로 IBM조종벡토르나 꼬리표 붙은 열쇠에 의하여 지원되지만 전통적인 조작체계보안에서 API들은 이 문제를 무시한다.

물론 열쇠관리를 특별히 간단하게 하는 보안대면부의 위치는 두 곳이다.

1. 열쇠관리는 대면부우에 있는 층에 전적으로 맡긴다. 아래층에는 요구하는 과제에 해당한 열쇠들이 주어 진다고 가정한다.
2. 열쇠관리는 대면부아래에 있는 층에 전적으로 맡긴다. 웃층에서는 그 어떤 열쇠관리조작도 하지 않는다.

주어 진 열쇠와 자료에 대하여 암호화조작을 실행하기만 하는 아주 간단한 암호화모듈(CM)은 첫번째 부류의 대면부를 통하여 접근될수 있다. 암호화모듈은 열쇠들이 잘못 쓰이지 않도록 하고 보안에 관한 자료의 인증을 담보하기 위하여서는 웃층에 의존하여야 한다.

응용프로그람작성자들이 보안전문가가 아니므로 보안관리에 될수록 적게 참여하여야 한다고 가정하면 두번째 부류의 대면부가 더 좋을것이다. 맨 웃준위의 대면부아래에 있는 층에서는 보다 많은 암호화알고리듬을 실현할수 있다.

그러면 GSS-API는 이 구성에서 어디에 적합한가? 단위통보문호출의 점에서 볼 때 GSS-API는 높은 준위의 대면부이다. 보안관리의 측면에서 GSS-API는 비교적 낮은 곳에만 있으나 그래도 높은 준위에 있다고 볼수 있다. GSS-API는 낮은 층에 의거하여 대부분의 열쇠관리자료들을 정의하고 보호한다. 그러나 일부 보안관리과제는 응용프로그람작성자에 의하여 실현되여야 하므로 호출자에게 그 권한을 대부분 남겨 두었다. 보안관리의 특징을 계속 증가시키면 GSS-API의 관리측에서 추상화준위가 이러저러하게 《약해 진다》는 위험이 있을수 있다. 실제로 GSS-API의 계속되는 공개판들에서는 관리호출의 수가 안전하게 증가되였다.

제4절. CORBA 보안

GSS-API는 보안봉사모임의 대면부이다. 분산체계에서 이 봉사층이야말로 보안시행을 위한 가장 적합한 위치로 된다.

첫째로 분산체계들은 각이한 조작체계와 다양한 하드웨어구성방식들을 결합하게 한다. 물론 조작체계설계자들이 보안시행에 대한 공통적인 규격화에 동의한다면 더 론의할 필요는 없다. 따라서 사용자들에게 균일한 보안대면부를 제시하기 위하여서는 그것을 조작체계우에 놓아야 한다.

둘째로 보안을 요구하는 응용쏘프트웨어를 고찰하자. 보안을 진행하기 위해서는 이 응용이 어떤 방법으로 조작체계와 호상작용하지 않으면 안된다. 응용프로그람작성자가 여러가지 다른 조작체계들을 대상하는 경우에는 성공전망이 다시 어두워 진다. 응용프로그람작성자들에게 밑에 놓이는 조작체계들의 대면부를 제공하는 중간층을 제시하면 해결될수 있다. 객체요청중개자(ORB)는 사용자와 객체들사이 또는 객체들사이의 호상작용을 조종한다.

아래와 우에로의 선택을 제한하면 그림 10-4에서 보여 준 해결방도에 도달할수 있다. 보안봉사들은 응용층과 조작체계의 사이에 있으면서 분산체계의 전체 구성요소들과 관계되는 공동층을 제공한다.

1. 객체요청중개자

만일 독자들이 보다 큰 어떤 조직의 IT체계를 조사할 기회를 가진다면 아마도 IT체계가 사용자들의 요구의 변화에 맞게 부단히 새로운 응용프로그람들을 첨부하면서 오랜

세월에 걸쳐 구축되였다는것을 알게 될것이다. 대부분 이러한 응용프로그람들은 기종이 다른 하드웨어와 쏘프트웨어기동환경(platform)을 대상으로 서로 다른 자료형식으로 각이한 언어로 작성되여 있으므로 응용프로그람들사이의 자료를 공유하는것은 결코 간단한 문제가 아니며 흔히 전용쏘프트웨어가 필요하게 된다. 자료가 기관들사이에서 공유된다면 보다 큰 규모에서도 같은 문제가 제기된다.

응용층		응용층
봉사		봉사
조작체계		조작체계
OS핵심부		OS핵심부
하드웨어		하드웨어

그림 10-4. 분산체계의 보안

분산객체콤퓨터작업은 이러한 환경을 해결하여 응용의 호상조작성을 개선하려고 시도하고 있다. 분산객체계산은 의뢰기-봉사기모형과 같은 분산콤퓨터작업에 대한 개념과 객체지향콤퓨터작업의 개념에 기초하여 구축되였다. 응용은 객체로 생각할수 있다. 객체는 속성과 방법을 가지며 객체의 속성은 객체의 방법으로만 조작될수 있다. 객체요청중개기는 대상들의 모든 통신을 조종하는 의뢰기와 봉사기객체들사이에 놓여 있는 《쏘프트웨어모선》이다. 공동객체중개기구성방식(Common Object Request Broker Architecture: CORBA)은 객체관리그룹(Object Management Group)에 의하여 개발되였으며 그러한 구성방식에 대한 공업규격화의 규격으로 되고 있다[61].

그림 10-5는 분산객체콤퓨터작업의 특징을 보여 준다. 응용은 의뢰기와 봉사기에 분산되여 있다. 의뢰기에서 대용체(stub)와 봉사기에서 골격(skelton)은 응용과 ORB서고사이의 대면부를 제공한다. 의뢰기의 응용이 봉사기의 방법을 호출하게 될 때 다음의 순서로 사건이 발생한다.

1. 의뢰기에서 방법호출은 자기의 응용대용체를 통하여 ORB기능을 불러 낸다.
2. 의뢰기에서 ORB서고는 활성화요소에 요구를 보내여 대상의 결합을 초기화한다.
3. 활성화요소는 방법호출에 의하여 주소화된 객체의 참조를 보관소로부터 검색한다.
4. 활성화요소는 봉사기에 있는 객체에 착수한다.
5. ORB서고들은 객체의 참조를 의뢰기에로 되돌린다.
6. 방법의 호출이 실행된다.

ORB서고에는 객체봉사가 들어 있다. 객체봉사의 제안된 목록을 아래에서 전부 보여 주었는데(원고를 작성할 때까지는 실현된것이 몇개에 불과하였다.) 여기서 보여 준 봉사이름들을 보고 그 기능을 충분히 알수 있으리라고 본다.

- 이름봉사(naming service)
- 사건봉사(event service)
- 거래봉사(transaction service)

- 지속객체봉사(persistence object service)
- 생명주기봉사(lifecycle service)
- 특허봉사(licensing service)
- 병행조종봉사(concurrency control service)
- 시간봉사(time service)
- 외부봉사(externalization service)
- 질문봉사(querying service)
- 관계봉사(relationship service)
- 속성봉사(property service)
- 보안봉사(security service)
- 무역봉사(trading service)
- 수집봉사(collection service)
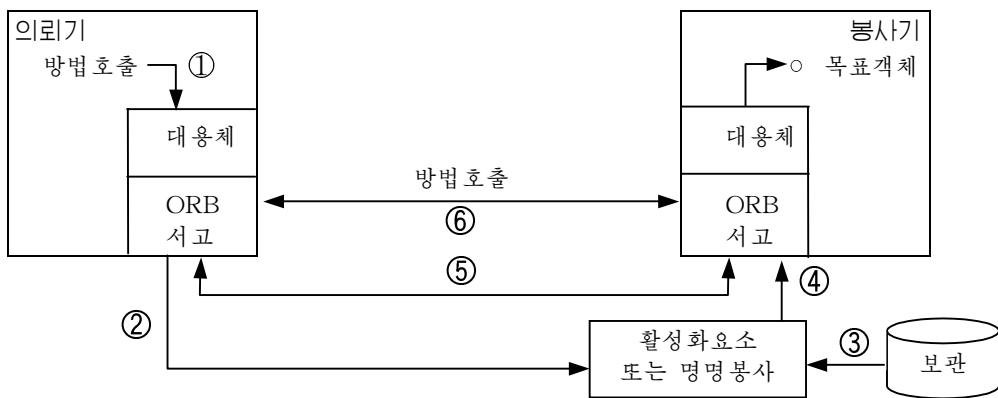
그림 10-5. CORBA의 방법호출

2. CORBA보안모형

CORBA보안구성방식설계자들이 대상으로 한 과제를 평가하기 위해서는 그 기본설계의 난문제들을 알아야 한다. 유연성과 호상조작성은 서로 상반되는 문제이다. CORBA는 폭 넓은 응용과 보안방책을 지원하는데 유연하여야 한다. CORBA는 또한 보호령역사이의 호상조작을 허용하는 틀거리도 제공하려고 한다. 호상조작성은 의견이 일치하여야 한다. 어떠한 형태의 규격화에 기초하여 동의가 이루어 져야 하는데 규격화는 그 본질상 유연성을 제한하고 있다. CORBA는 보안봉사를 실현하고 관리하기 위한 틀거리를 제공하지만 개별적인 보안기구들을 규정하지는 못한다.

CORBA보안은 부단히 변화되는 환경에서 관리되여야 한다. 분산객체들은 언제나 창조, 삭제, 변경되므로 객체들사이의 호상작용은 복잡할수 있다. 일부 응용들에서는 자기에게 필요한 보호를 명백히 요구하기때문에 보안에 빈틈이 없지만 반대로 보안에 빈틈이 있는 응용에서는 어떤 수단을 써서라도 응용과 련관시키지 않고 보안을 실현하여야 한다. 봉사를 요구할 때 객체들은 특권을 위임하여야 하므로 방책들은 접근권한의 위임

한계를 제한할수 있다. 실례로 어떤 방책에서는 목표객체의 특권을 그의 유효기간이나 특권이 호출될수 있는 회수에 위임할수도 있다고 제정하고 있다.

보호에 대한 요구가 비슷한 객체들은 령역으로 분류된다. 령역보안방책은 ORB에 의하여 리행된다. CORBA에는 접근조종은 물론 방책객체를 관리하기 위한 대면부들이 포함되여 있다. CORBA는 특정의 방책들을 대상으로 하지 않는다.

ORB사이의 호상조작성은 다리, 관문 그리고 General-Inter-ORB Protocol (GIOP)과 Internet Inter-ORB Protocol(IIOP)에서처럼 내부ORB규약들에 의하여 공학적준위에서 지원될것이다. SECIOP는 보안내부ORB규약이다. 호상조작성문제는 취급하기 힘들므로 이 책을 쓸 때까지는 실제적으로 실현된 ORB다리는 없었다. 호상조작성은 또한 령역사이의 방책이 일치할것을 요구하고 있다. 다음과 같은 표준적인 접근권한의 모임이 있다.

- get(얻기)
- set(설정)
- manage(관리)

유연성을 위하여 추가적인 권한을 정의하기 위한 선택항목도 있다. CORBA보안봉사에 다음과 같은것들이 들어 있다.

- 인증,
- 보안문맥설정,
- 인증화와 호출조종: ACL, 능력, 역할에 기초한 접근조종,
- 통보문보호:주로 암호화방법으로 실현하지만 다른 방법으로도 할수 있다.
- 검열,
- 비참가거부.

이제부터 우의 봉사들에서 처음 두가지만을 간단히 고찰하자.

3. 인증

그림 10-6은 CORBA에서 인증방법을 보여 주고 있다. 처음에 사용자는 ORB에 서명한다. 사용자주최자 실례로 등록가입프로그람은 사용자신분과 통과암호를 당사자인증기객체에 넘겨 준다(걸음 1). 그러면 당사자인증기객체는 신임장객체를 창조한다(걸음 2). 이 신임장에는 인증된 신분, 역할, 특권과 같은 사용자보안속성들이 들어 있다. 그 다음에 사용자주최자는 신임장객체의 참조를 현재의 실행문맥을 반영하는 Current객체에 넘겨 준다(걸음 3). 이제부터 사용자는 봉사를 호출하려고 한다(걸음 4).

보안봉사들은 의뢰기객체와 봉사기객체사이 접근을 중개한다. 먼저 의뢰기와 봉사기사이에 보안련계(결합)를 설정해 주어야 한다. 매개의 보안련계에서 의뢰기와 봉사기쪽에 Security Context객체가 있다. 보안문맥에는 보통 객체가 리용하게 되는 암호화알고리듬과 암호화열쇠의 속성이 있다.

보안호출봉사는 이러한 련계가 이미 존재하는가를 검사하고 만일 보안련계를 새롭게 설정하여야 하는 경우에는 도약봉사를 호출한다. Current객체로부터 오는 보안정보는 해당한 속성을 얻는 과정에 쓰이게 될것이다(걸음 5).

보안봉사와 그 작용은 보안을 모르는 응용으로부터 완전히 은폐된다.
설정된 보안련계는 기타 보안봉사 실례로 접근조종에 의하여 리용될수 있다.
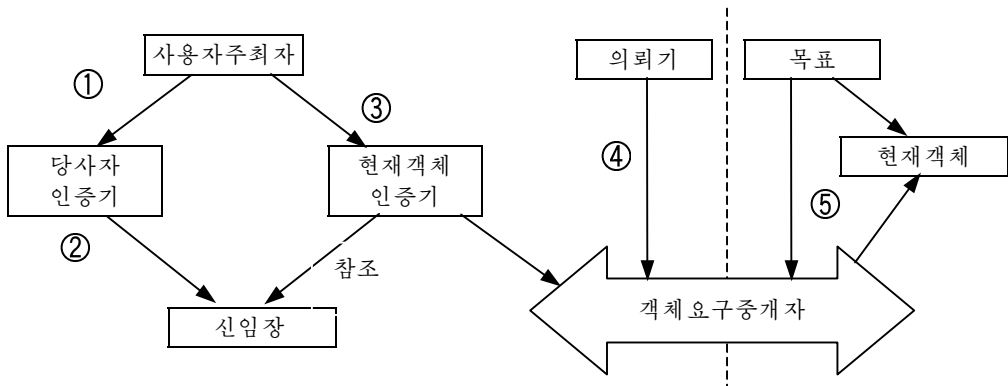
그림 10-6. CORBA에서의 인증

4. 보안이 담보되는가 아니면 보안봉사가 담보되는가

CORBA는 보안을 담보한다고 약속하고 있지만 이것이 준수되고 있는가? 사실여부를 검토해 보자. CORBA테두리에서 보안은 응용에 맡겨 두지 않았으므로 모든 요구들은 ORB를 통과하여야 하며 ORB는 보안조종을 응용한다. 만일 보안봉사들이 모두 적당하게 설정되여 있고 사용자들은 ORB를 우회할 길이 없고 즉 조작체계를 간접적으로 호출한다면 CORBA는 보안을 완전히 담보한다.

그러나 CORBA는 ORB를 우회할수 없으며 CORBA의 보안봉사기에 의하여 리용되는 자료가 적당히 보호된다는 담보는 없다. TCB는 보안에 관계되는 정보들 실례로 암호열쇠나 신임장들을 언제나 보관하는 모든 기계들에 대한 조작체계와 하드웨어들을 포함하고 있다. 이외에도 CORBA에서 보안은 봉사에서 쓰이는 구체적인 보안기구의 강도에 의존한다. 이러한 체계들에서 강도가 조금이라도 약하면 ORB아래에 있는 층으로 호출이 허용되므로 CORBA보안을 약화시킬수 있다. 이런 의미에서 CORBA는 결코 그 어떤 담보도 주지 못한다고 말할수 있다. 보안의 틀거리로부터 더는 기대를 가질수 없고 또 기대하지도 말아야 한다는것이 공명정대한 견해로 된다.

CORBA는 분산대상체계에 대한 명세이다. 조직들은 이제야 CORBA와 CORBA보안의 실현을 위한 조사에 겨우 착수했을뿐이다. CORBA의 계층화모형에서 올라 갈수록 보안기구를 완벽하게 실현하는것은 더 힘들어 진다. 바로 그렇기때문에 제작자들은 아주 복잡한 과제에 맞다들리게 되며 CORBA 보안기능을 더욱 높이기 위하여 노력하게 될것이다. CORBA보안이 실제적으로 어느 정도로 담보를 줄수 있겠는가 하는 경험은 아직 없다.

이 장의 문헌안내

[83,159]는 분산체계에서의 인증리론과 실천을 구체으로 고찰한 론문이다.커베로스의 구체적인 사항에 대해서는 Internet RFC 1510[80]에서 고찰하였다.커베로스보안의 해석과정과 개발된 환경을 [15] 에서 주었다. 커베로스에로 확장할 때 Privilege Attribute Cerrificats (PAC)에 의한 자기의 접근조종특징들을 SESAME[123] 또는 OSF DCE에서 구체적으로 론의되였다.

크립토나이트(Kryptoknight)는 IBM이 커베로스에 대응하여 개발한 체계이다. 커베로스와 마찬가지로 크립토나이트는 보안봉사가 인증과 열쇠분배를 제공하는 《중심체계》이다. 크립토나이트에서 암호화규약은 암호화대신에 반출문제(export problem)를 피하기 위하여 완정성검사기능들을 리용하고 있다[20].

보안규약이 중심보안봉사기에 의존하며 이 봉사기에 응용될수 없다면 호출조종을 결정하기 힘들다. 그러므로 호라스보안구성방식(Horus Security Architecture)에서는 오유허용한계를 분산보안체계에 받아 들였다. 보증서분배쎈터우에서 단일하여야 한다는 엄격한 직결처리의 요구를 피하기 위하여 여러가지 봉사기들의 증명서를 분배하는데 비밀을 공유하는 방안이 쓰인다[126].

GSS-API는 1993년 9월에 Internet RFC 1509 로써 공개되였다. 수정판 2는 1997년 1월에 RFC 2078로 공개되였다[88].

CORBA에 대한 기초도서들과 구체적인 사항, 론문들은 다음의 주소에서 찾으면 된다.

http://ww.omg.org
http://www.acl.lanl.gov/CORBA

[106]과 [121]를 참고하면 CORBA 들에 대한 충분한 리해를 가질수 있다.

련습문제

1. 사용자신원에 기준한 접근조종방책들은 분산체계의 로화를 앞당긴다. 그래도 분산보안체계에서의 접근조종방책들은 사용자신원에 기초하여야 하는가? 이러한 방책들과 다른 방책을 찾아 보시오. 분산체계에서 개별적인 사용자들은 언제 인증되는가?

2. Lattic_1로부터 Lattic_2에로 그리고 그 반대로 표식들을 변환하는 변환함수 T_1와 T_2을 정의하시오(그림 10-7). Lattice_1을 리용하는 체계로부터 오는 주동체 또는객체들이 Lattice_2를 리용하는 체계에로 전송되며 따라서 아래방향에로의 정보흐름이 허용되지 않는 경우에는 이 함수들이 적용될것이다. 변환함수를 선택하기 위하여 항목들이 Lattice_1로부터 Lattice_2에로 전달되였다가 다시 Lattice_1로 되돌아 왔다

면 항목들에 어떤 표식자가 붙게 될것인가를 보여 주는 표를 구성하시오(수학적인 용어로 표현하면 련결 $T_2 \circ T_1$를 계산하시오). Lattice_2 에서 초기에 표식자가 붙은 항목과 같게 하시오.
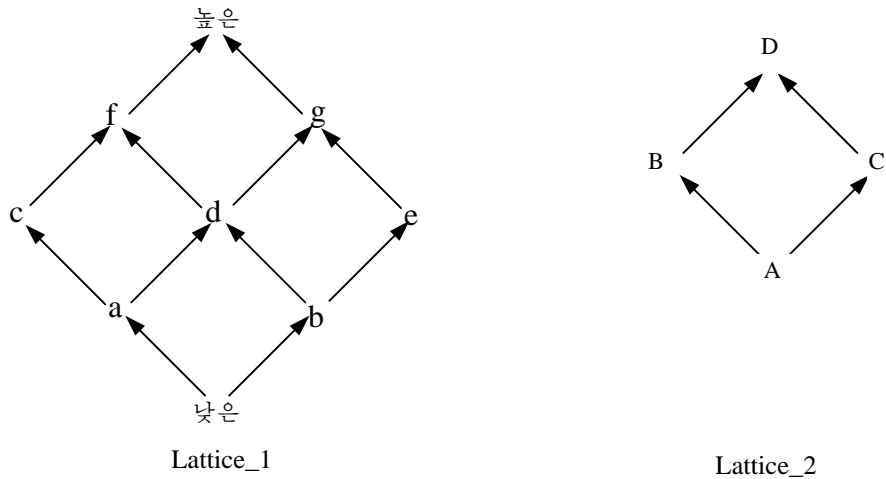
그림 10-7

3. 크립토나이트(KryptoKnight)규약을 간단히 서술하고 커베로스와 비교하여 그의 우점과 결함을 이야기하시오 .

4. 령역들사이의 접근을 허용하는 확장된 커베로스를 설계하시오. 이러한 방안이 쉽게 실현되자면 어떠한 관리적배렬이 있어야 하는가? 어떤 걸음들을 규약에 첨부해 주어야 하겠는가?

5. 보증서들은 보증취소목록 또는 직결식취소봉사기들을 통하여 취소될수 있다. 사용자들이 자기의 증명서를 보관하는 개인용암호주변장치를 휴대하는 분산체계에서는 어느 방법이 더 좋다고 말할수 있는가?

6. GSS-API를 리용하여 응용을 실행시키려고 할 때 자기의 체계에 어떠한 쏘프트웨어 요소들을 보충해 주어야 하겠는가?

7. 분산체계에서 보안의 국부적인 시행을 반대하는 론거로는 어떠한 원인들을 들수 있는가?

제11장. WWW보안

WWW는 분산콤퓨터도입을 새로운 수준으로 올려 세웠다. 이동가능한 코드는 인터네트를 통하여 의뢰기에로 이동하여 의뢰기우에서 실행된다. 전자상업은 새로운 업무가능성을 약속하고 있다. 뜻밖에도 보안은 IT분야에서 주되는 관심사로 되였다. 따라서 IT체계를 리용하고 있는 방법에서 상당한 변화가 있는 정황속에 있는 자신을 다시한번 발견하게 된다. 다음과 같은 물음들이 제기된다.

- 낡은 보안구성방식들이 계속 적합한가 아니면 새로운 방책들과 새로운 시행기구들이 필요한가?
- 보안을 고려하지 않고 설계되였던 IT체계에서 보안의 토대를 어떻게 마련할수 있겠는가?

이 장에서는 WWW보안에서 제기되는 주요문제들을 간단히 고찰한다.

목적

- Web콤퓨터작업의 일련의 측면들이 어떻게 되여 새로운 보호요구로 되였는가를 인식한다.
- 다종다양한 보호문제가 있다는것을 리해한다.
- 오늘날 이 문제들을 해결하는데 쓰이는 기구들을 개괄한다.
- 지적소유권의 보호를 둘러 싸고 있는 문제점들을 간단히 고찰한다.

제1절. 배경

WWW는 규모가 방대해 져 압도적인 수를 이루는 보안문맹자들이 새로운 콤퓨터기술과 직접적으로 접촉할수 있게 하였다.초기에 분산체계응용은 의뢰기-봉사기모형에 기초하였다. 의뢰기가 봉사기에서 계산을 진행하려고 하면 봉사기는 자신을 보호하기 위하여 의뢰기를 인증한다. 이러한 방법으로 제공되는 봉사의 수가 그리 많지 않았기때문에 봉사에서 보안의 결함을 결과적으로 제거해 버릴수 있게 되였다.

정황은 어떻게 변화되였는가? 초기에 WWW는 본문, 도형, 음성 등이 들어 있는 하이퍼본문문서를 창조하고 처리하기 위한 규격화를 제공하였다. 하이퍼본문문서(hypertext document)를 작성하기 위한 초기의 규격화가 HTML(HyperText Mark-up Language)언어이라면 대응하는 전송규약은 HTTP(HyperText Transfer Protocol, Internel RFC1945)이다. 하이퍼본문의 특징들은 사용자들이 정보(내용)를 검색하게 하는 새로운 응용에서는 인기가 있었다. 신문란, 려행시간표 또는 려행안내도와 같은 려행정보, 회의공시, 그림, 음향, 쏘프트웨어, 기술문서 그밖의 다른것들이 내용으로 될수 있다. 최근에는 날로 늘어 나는 상업봉사제공자들이 이 시장에로 뛰여들고 있다. WWW는 불가피하게 분산콤퓨터화의 기본성질을 변화시켰다.

- 여기서는 프로그람과 자료의 분리를 하지 않는다. 내용제공자들은 문건에 실행가능내용(애플레트)을 매몰함으로써 사용자입력을 처리할수 있는 쌍방향리용이 가능한 Web페지를 창조하였다.
- 콤퓨터작업은 의뢰기에로 넘겨 진다. 문서들에는 실행가능한 코드가 들어 있고 의뢰기들은 대단히 성능이 높은 기계들에서 동작하므로 봉사기들은 콤퓨터작업과제를 자기의 의뢰기에 넘겨 주고 자원을 해방할수 있다. 때문에 의뢰기는 나쁜 속심을 가진 내용제공자들로부터 보호할 필요가 있다.
- 이동코드들은 여러 위치들로부터 정보를 수집하거나 희귀한 계산자원을 찾으면서 기계에서 기계에로 이동한다. 의뢰기는 이동코드로부터 보호하여야 한다. 이와 반대로 이동코드들이 동작하고 있는 의뢰기에서 이동코드를 보호할 필요도 있을수 있다.
- 사용자들은 체계관리자와 방책작성자로 되지 않으면 안된다.

Web는 또한 쏘프트웨어분산을 위한 새로운 모형도 창조하였다. 쏘프트웨어는 인터네트로부터 제공 받을수 있는 다른 부류의 내용으로 된다. 이 모형이 좋다고 말할수 있는 근거는 여러가지로 볼수 있다. 독자들이 이미 쏘프트웨어제공자들에게 련결되여 있다면 무엇때문에 플로피디스크에 있는 쏘프트웨어를 구입하겠는가? 그러나 초기 PC시대에서 얻은 경험을 상기해 보면 독자들은 아연해 질수 있다. 플로피디스크에 자원을 보관하게 한것은 콤퓨터비루스전염을 류포시키는 결과를 초래하였다. 많은 기관들이 플로피디스크들을 다른 기관들로부터 가져다 쓰는것을 각성해야 한다는것을 적지 않은 고충끝에 알게 되였으며 이것은 WWW에서도 받아 들이게 되였다.

WWW는 기본적으로 새로운 보안문제들을 만들어 내지 않았지만 WWW보안이 독자적으로 한개 장을 이룰 정도로 그 내용을 변화시켰다. WWW보안은 급속히 변화되고 있으므로 이 책이 완성되였다고 또는 최신으로 된다고 말할수 없다. 이 책의 목적은 지금까지 제안된 보안기구들을 설명하고 이러한 해결방법이 가지고 있는 고유한 제한성을 언급하는데 있다. 이 책에서는 콤퓨터보안이 기본문제로 되고 있으므로 인터네트에서 전송되는 자료의 보호는 간단히 언급만 한다.

제2절. Web열람기

WWW를 호출하기 위해서는 의뢰기에 Web열람기가 있어야 한다. Web열람기는 단순하게 말하면 사용자들에게 도형사용자대면부(GUI)를 제공하며 Web를 련결시키는데 필요한 규약들이 들어 있는 프로그람이다. 가장 많이 보게 되는 열람기들은 Netscape’s Navigator와 Microsoft Internet Explorer들이다. 열람기는

- 주목하는 Web페지를 나타내는 《종소리와 휘파람소리》를 발생하며
- Web응용을 위한 봉사층이며
- Web봉사기들과 통신을 하기 위한 규약들이 들어 있고
- 의뢰기의 보호에 관계되는 정보들을 관리한다.

의 특성이 있다. 여기서 고찰하는 Web보안모형의 기본요소들은 의뢰기,의뢰기열람기, Web봉사기이다. 이 장에서 Web봉사기라는 말은 봉사프로그람을 동작시키고 Web페

지를 관리하는 기계보다도 봉사를 제공하는 쏘프트웨어라는 뜻으로 더 많이 쓰이고 있다. 열람기는 의뢰기의 보안에서 중요하므로 현재의 제품의 고유한 특징과 열람기라는 개념에 고유한 일반적특징들을 구별해 보아야 한다. 열람기는 다음의 리유들로 하여 TCB의 한부분으로 되고 있다.

- 열람기는 의뢰기Web통신량을 조종한다. 의뢰기와 봉사기들사이에 원활하게 내용을 전송하기 위하여 열람기는 사용자와 사용자의 콤퓨터환경에 대한 정보를 봉사기에 《로출시킨다》. 최소한 열람기는 되돌림주소를 알려 주어야 한다. 이때 봉사기가 자기의 의뢰기들에 대한 자료기지를 구축하고 의뢰기들이 식별하지 못하게 그것을 리용한다면 사적비밀보호에 대한 문제가 제기될수 있다.
- 열람기들은 의뢰기의 환경에 대한 기정설정값들과 선택값들을 관리한다. 기정설정값들에는 실행가능한 위치가 들어 있다.보안의 선택값들은 보호의뢰기들이 Web 대화를 응용하려고 한다는것을 나타내고 있다.
- 열람기들은 최근에 방문한 페지들의 경력과 완충기억을 보관하고 있다. 이것은 사용자들에게 편리하다. 국부적인 완충기억에서 검색할수 있는 페지로 가면 속도를 높일수 있다. 이제 공동말단의 실례로 비행장대합실에서 려행자들에게 Web봉사를 제공하는 경우를 생각해 보자. 많은 려행자들이 이 말단을 차례로 리용한다. 이전 페지들에로 넘어 간다는것은 다른 려행자들이 방문한 페지들에로 돌아 간다는것을 의미한다. 안전한 Web열람기는 객체재리용문제를 취급하여야 한다 !
- Web보안응용들에서는 암호화 및 수자서명알고리듬을 리용하는것이 좋다. 열람기가 의뢰기에 대하여 우의 알고리듬을 수행하는 경우 열람기에 의뢰기의 비공개열쇠를 의탁하여야 한다. 들어 오는 통신량과 보증서에 대한 수자식서명들은 검사되여야 하므로 오늘날 열람기는 중요보증실체의 뿌리대조열쇠를 가지게 된다. 명백한것은 열람기가 변경과 서명으로부터 검증열쇠를 보호하여야 하며 또한 로출로부터 암호열쇠를 보호하여야 한다는것이다.
- 사용자들에게 인터네트접근을 위한 유일한 도구를 제공하기 위하여 열람기들은 전자우편과 같은 다른 통신봉사들을 통합한다. 보안의 각도에서 볼 때 이것은 불필요한 복합프로그람을 리용하는것으로 된다. 공격자는 열람기의 바그들을 리용하는 전자우편통보문들을 보낼수도 있다. 초기의 우편프로그람은 이 공격으로부터 면역성을 가지게 할수도 있었는데 그렇게 하지 못하였다. 봉사들을 통합하면 통합하지 않았을 때 예견하지 못했던 호상작용이 발생할수 있다.
- 열람기는 보통 전체 체계자원에 대하여 완전한 접근을 가지는 체계방식에서 동작한다.
- 총체적으로 볼 때 초기의 조작체계에 의하여 실행되던 기능들이 더욱더 풍부해 진다고 보면 열람기들은 이를테면 Microsoft Internet Explorer 4 에서와 같이 일정한 시기에서는 조작체계의 통합부문으로 될수도 있다. 이 과정에 열람기들은 열람기자체의 접근조종 또는 어떤 Web페지의 접근조종을 하는 사용자인증과 같이 보안의 사명을 수행하게 될것이다.

열람기들이 상품화되지만 그의 내부명세들이 공개되지 않는다면 사태는 복잡해 지게 된다. 전문가들까지도 때때로 지나친 고찰방법이 필요 없다고 인정하게 된다[128].

제3절. CGI스크립트

전통적인 의뢰기-봉사기모형에서 여러개의 의뢰기들이 접근하는 봉사는 FTD와 같은 불과 몇개밖에 되지 않으며 원격수속접근(RPC)을 통하여 광대역호출을 하는 의뢰기의 수도 적다. CGI 스크립트는 많은 의뢰기들에 대하여 보다 유연하게 봉사에 접근할수 있게 한다. 보안문제의 본질은 변화되지 않았다. 봉사기는 자기의 의뢰기들에 대한 조종된 호출을 제공한다. 그러나 문제의 폭은 다르다. 많은 사람들에 의하여 작성되고 보다 많은 일을 할수 있는 프로그람들을 더많이 동작시킬것을 봉사기에 요구한다. 매 항목 그자체가 보안의 관심사로 되여야 한다..

CGI(Common Gateway Interface공통관문대면부)는 URL(Uniform Resource Locators)들이나 HTML을 실행가능한 프로그람으로 변환하기 위한 메타언어이다. 이 프로그람들을 CGI의 규격화요구가 적은 언어들에서는 다 작성할수 있다. Perl, Tcl 또는 Safe-Tcl과 같은 스크립트언어들이 여기에 특별히 적합하다고 볼수 있다. 보다 일반적인 의미에서 CGI는 의뢰기들이 봉사기에게 실행할것을 요구하는 계산에 영향을 미치는 선택권을 의뢰기에게 더많이 준다는 총체적인 개념을 의미한다.

CGI는 다음과 같이 동작한다(그림 11-1). CGI스크립트와 그의 입력변수를 기입하면 의뢰기는 URL이나 HTML의 형식으로 봉사기에 보낸다. 이 요구는 Web봉사프로그람의 사용자신분에 의하여 실행되는 어떤 프로그람에 전송된다. Web봉사기는 봉사기측 포함(Server-Side Includes, SSI)과 같은 응용프로그람을 호출할수도 있다. 이 봉사기측 포함이라는 응용프로그람에 의하여 봉사기우의 문서는 SSI즉시처리라고 부르는 체계지령들을 포함할수 있다. 어떤 의뢰기가 이러한 문서를 요구하면 이 체계지령들은 평가되여 그 결과를 문서에 삽입하고 의뢰기에로 되돌려 준다.

[128]의 간단한 실례를 통하여 CGI스크립트들이 어떻게 손상을 일으키는가를 알수 있다. 파일을 의뢰기에 보내는 스크립트는 다음과 같다.

cat thefile | **mail** client address

여기서 thefile은 파일의 이름이며 clientaddress는 의뢰기의 우편주소를 나타낸다.

나쁜 마음을 가진 사용자가
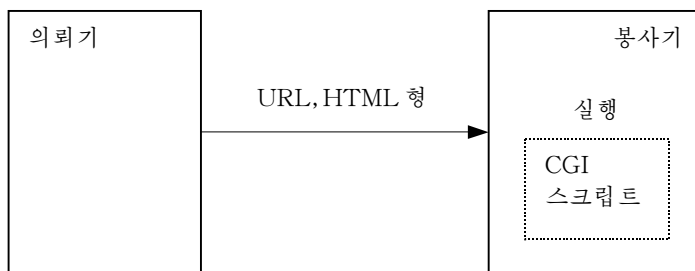
그림 11-1. 봉사기는 CGI스크립트를 실행한다

user @ address | rm-rf/

을 우편주소로 들여 보내면 봉사기는

cat thefile |**mail** user @ address | rm-rf/

을 실행할것이다. 파일을 사용자에게 우편으로 보낸 다음 지우기를 허용하는 모든 파일들을 삭제한다.

Web봉사기체계에서 보안을 관리하기 위하여서는 많은 과제들이 실행되여야 한다. 아래에서 체계가 Unix체계라고 가정한다. 먼저 체계관리자는 CGI스크립트들을 주체계우에서 추적하여야 한다. 다음의 두가지 선택권이 있다.

- **Script-aliased CGI**: 모든 CGI스크립트들은 하나의 Web봉사기뿌리등록부 즉 **/var/httpd**의 어떤 등록부 **./cgi-bin.**에 넣는다.
- **Non-script-aliased CGI** : 모든 CGI스크립트들은 자기의 확장자 **.cgi**에 의하여 식별된다.

첫번째 선택권은 모든 CGI스크립트들을 보다 쉽게 찾아 내므로 보다 《안전》하다. 다음에 UID우에서 Web봉사기프로그람을 결정하여야 한다. 최악의 경우에도 대처할수 있어야 하므로 CGI스크립트가 의심스럽다면 조종을 하지 않을수 있다. Web봉사기프로그람을 뿌리로서 동작시키면 재난을 초래할수 있다. 최적이라고 볼수 있는 선택권은 전용적인 Web봉사기 UID를 창조하고 그의 접근권한을 충분히 고려하여 조종하는것이다. 이 해결책이 서로 다른 사용자들에 속하는 Web페지들을 분리시키지 않는다는것을 명심하여야 한다. 모든 CGI스크립트들은 동일한 UID에서 동작할것이다. 스크립트가 자기의 저자의 허가밑에서 동작하도록 하기 위해서는 CGI Wrap와 같은 포장프로그람이 필요하다.

Web봉사기의 통합환경을 보호하기 위해서는 Web봉사기 UID가 Web봉사기2진형파일과 구성파일을 가지지 말아야 하며 또한 Web봉사기 UID는 다른 봉사들과 공유되여서는 안된다. 특히 Web봉사기는 UID-2를 가진 특수한 사용자 Nobody에서는 동작시키지 말아야 한다. UID환경에서는 다른 봉사가 이미 동작하고 있다는것을 예견하여야 한다.

CGI스크립트의 코드를 다시 조사해 보면 보안에서 결함이 있는 스크립트를 제거할수 있다. 만일 시간과 풍부한 경험이 있어 이것을 한번 해본다면 이것은 더 말할것없이 좋은 생각이다. 될수록 빨리 봉사기우에서 자기의 새로운 페지를 보려고 하는 가입자들의 Web싸이트를 관리하려고 한다면 이 선택권을 쓰지 않아도 된다.

끝으로 호출을 조종하는 다른 경우와 마찬가지로 CGI스크립트의 입력을 려파하지 않으면 안된다. 호출되는 조작이 유력할수록 수신하는 입력에 보다 깊은 주의를 돌려야 한다. 봉사기측의 포함프로그람은 기능이 매우 풍부하다. SSI즉시처리형의 형식은 다음과 같다.

<!-#operator arg1=" string1" arg2=" string2" …->

인수 cmd를 가지는 연산자 exec 에 의하여 최종적으로 유연성이 보장된다.
SSI즉시처리형의 형식

<!-#exec cmd=" myprogram myparameters" ->

은 문자렬 myprogram myparameter를 실행을 위하여 /bin/sh에 넘겨 준다. myparameters에 쉘확장문자가 들어 있는 경우 완전히 해를 주지 않는 프로그람에 넘겨 진 파라메터 또는 프로그람으로부터 범죄가 발생할수 있다. “unescape”조작은 확장문자들에 주석을 달아 줌으로써 의뢰기에서 오는 입력에서 쉘확장문자를 제거한다. 지령

unescape “string1; string2”

은 “string1\ ; string2\ ”을 되돌린다. 스크립트언어(실례로 Perl)들도 확장문자를 해제한다. 확장문자를 해제하면 봉사기측의 포함프로그람에서 제일 공격하기 쉬운 구멍이 없어 지게 된다. 공격자들은 여전히 Unix 의 지령들과 자기들의 입력으로 공격하고 있다. 이러한 모험을 접수할 준비가 되여 있지 않다면 다음의 봉사기선택권으로서 exec를 해제시키면 된다.

Option IncludeNOEXEC

제4절. 쿠키

어떤 형태의 업무에서든지 개별적인 의뢰기를 마음대로 선택하도록 봉사를 변경시킬수 있다. Web봉사도 례외가 되지 않는다. Web봉사인 경우에는 가입자들에 대한 정보를 기억하는 장소가 필요하다. 이 정보는 봉사기에 보관될수도 있지만 사용자기지가 방대한 경우에는 요구되는 기억기요구와 탐색시간이 문제로 된다. 더우기 HTTP요구들은 개별적인 사용자들을 자동적으로 식별하지 못하므로 협동열람기의 도움으로 가입자싸이트를 리용하는것이 보다 쉬워 진다. 봉사기는 열람기에 봉사기가 다음번 의뢰기호출에서 참조할 정보를 포함하는 쿠키(cookie)를 보관하여 두도록 한다(그림 11-2).Unix체계에서 대표적인 기억위치는 사용자홈등록부에 있는 .netscape/cookies와 같은 파일이다.
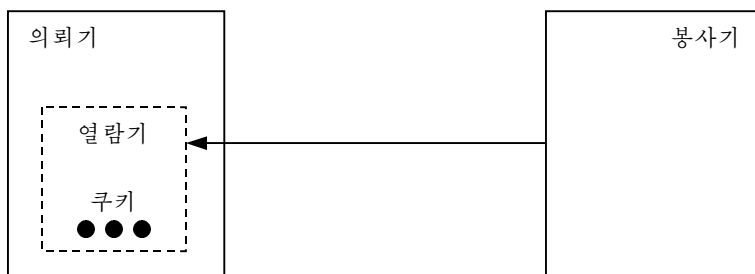
그림 11-2. 의뢰기에 보관된 쿠키들

쿠키를 사용하는 공학적인 리유도 있다. HTTP규약은 국적이 없다. 모든 HTTP요구들은 독립적인 사건으로 취급되며 지어는 동일한 의뢰기로부터 온다고 해도 마찬가지이다. 련관되여 있는 모든 관리과제들은 부단히 반복된다. 실례로 Web페지에 접근하기 위한 통과암호가 요구된다면 이 페지에서 마우스를 찰칵할 때마다 이 통과암호를 돌려 주어야 한다. 이 문제는 HTTP 1.0에 의하여 대화를 진행하는 기간에 이미

해결되였다. 열람기는 첫 요구에서 들어 온 통과암호를 기억하고 이것을 이후의 봉사기에 대한 모든 응답들에 자동적으로 포함시킨다.

쿠키들은 이 개념을 일반화하여 사용자간접소비시간은 물론 행정적인 간접소비시간을 줄이면서 국적 있는 HTTP 대화를 창조할 가능성을 열람기들한테 준다. 그러면 국적정보는 대화기간이 지나도 보관될수 있다.

쿠키들은 봉사기로부터 의뢰기에로 작업부담을 옮기는 잠정적인 첫 단계이다. 쿠키들이 보안문제로 되는가? 쿠키들은 체계의 통합환경을 위반할수 없다. 그것은 쿠키들이 실행가능한 코드가 아니라 자료이기때문이다. 쿠키들은 봉사기에 직접적으로 정보를 로출시키지 않는다.결국 봉사기는 열람기에 쿠키를 보관하라고 요구한다. 개별적인 쿠키들도 역시 어떠한 기밀성문제를 만들어 내지 못한다.

사적비밀문제가 남아 있으므로 개별적인 쿠키들에 근심할 필요가 없다. 어쨌든 봉사기들은 저마다 정보를 얻는다.그러나 열람기에 의하여 기억된 쿠키들의 전체 모임은 의뢰기프로필을 창조한다. 따라서 열람기접근조종기능은 매우 중요하다. 일반적으로 쿠키들은 령역에 특정이며 봉사기들한테는 자기 령역에 속하는 쿠키에 대한 접근만이 주어 진다. 최근에 대부분의 열람기들은 쿠키들을 기억시킬수 있는가를 물어 보고 설정할수 있는데 이것은 시끄러운 문제로 되기가 쉽다. 쿠키들을 기억하지 않는 열람기들도 있으며 이 경우 대화의 끝에서 쿠키들을 지우는 선택권이 반드시 있게 된다.

제5절. 보증코드

프로그람은 저자에 의하여 서명되며 의뢰기는 봉사기로부터 얻어 지는 프로그람에 있는 수자식서명을 검증한다(그림 11-3). 따라서 의뢰기는 코드원천 또는 일반적으로 내용원천을 검증할수 있다. 이전에는 류사한 개념으로서 수축포장된 쏘프트웨어(shrink-wrapped software)가 있었다. 보증서는 콤퓨터보안문제보다도 통신보안문제를 상대로 한다. 인터네트에 제품을 제공하는 쏘프트웨어작성자들은 다른 동업자들이 모방하지 못하도록 보호된다. 의뢰기들은 그들이 내리적재하려고 하는 코드원천을 아는 경우에 어떤 보호를 받을수 있다.
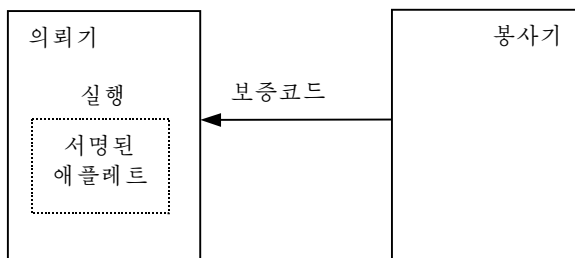
그림 11-3. 의뢰기에서 서명한 애플레트의 실행

이 방안에서 의뢰기들은 업무를 진행하려고 하는 봉사기들에 대하여 검증열쇠를 필요로 하고 있다. 의뢰기는 인터네트가 아니라 통로를 통하여 검증열쇠를 얻을수 있지만 검증열쇠대신에 보증서를 리용하는 경우가 많다. 이때 보증서는 다른 그 어떤 사람에 의하여 서명되여야 하며 의뢰기에서는 보증서를 검사하기 위하여 검증열쇠가 필

요하다. 자기의 가입자들에게 보증봉사를 제공하는 회사들이 존재한다. 이러한 방안밑에서 보증된 열쇠는 다 보증서의 사슬에 의해 검증될수 있는데 이때 사슬의 마지막요소는 이 방안의 뿌리서명열쇠밑에서 서명된다. 보증코드의 검증을 초기에 진행하기 위하여 의뢰기는 해당한 뿌리검증열쇠들을 가지고 있어야 한다. 오늘날의 열람기들은 이러한 열쇠를 가지고 있다.

해결되여야 할 문제가 아직 하나 남아 있다. 만일 서명을 검사하기전에는 인터네트에서 얻은 내용을 믿을수 없다면 그리고 인터네트로부터 자기의 열람기를 받는다면 어떻게 대조를 시작할수 있겠는가? 안전하자면 Web밖에서 뿌리검증열쇠의 유효성을 확증해야 한다. Web보안에 관한 문헌이 뿌리검증열쇠를 위한 아주 보편적인 원천으로 될수 있다.

보증코드는 내용이 어디서 오는가를 사용자들이 알고 있다고 담보한다. 보증서는 코드의 동작에 대해서는 그 어떤 담보도 주지 못한다. 명망이 있는 쏘프트웨어판매자들까지도 가끔 콤퓨터비루스를 전파하는 불량한 쏘프트웨어를 류포시킬수 있다는것을 잊지 말아야 한다. 사용자가 동의한 방안하에서 보증되지 않은 Web싸이트로 갈 때 보증코드들은 아무런 도움도 주지 못한다.

의뢰기는 자기가 믿는 원천에 대한 보증열쇠목록을 보관하고 있다. 이 목록은 명백히 공격대상으로 된다. 만일 해독적인 Web봉사기가 이 목록에서 자기의 검증열쇠를 꺼낸다면 이 봉사기로부터 오는 코드는 믿음성 있는 코드로 취급할수 있을것이다.

Active X조종에 쓰이는 Microsoft의 인증코드는 확증코드의 기본실례로 된다. Active X조종은 Web페지에 첨부되는 쏘프트웨어요소들이다. Active X의 조종이 확증되였다면 실행이 가능한 동작들을 더이상 제한하지 않고도 실행시킬수 있다. Microsoft와 Microsoft가 인정한 제공자들로부터 쏘프트웨어를 배포 받는것은 Web보안기구의 론리적인 방도로 된다. 이러한 방안에 근거하여 Web싸이트로부터 코드를 받으면 《담보된》 제공자로부터 프로그람을 사는것과 완전히 같다. 잘 알려 지지 않은 제공자로부터 받은 실행가능한 내용을 동작시키려고 하는 경우에 보증코드는 도움을 주지 못한다.

제6절. 모래통

Web기술의 능력을 충분히 발휘하기 위하여 사용자들은 관심을 가지는 어떤 Web싸이트로부터 실행가능한 내용을 접수할수 있도록 준비되여 있어야 한다. 이를 위해서 사용자들은 실행가능한 내용(애플레트)의 동작을 조종할수 있어야 한다. 이것은 바로 요구하는 환경에서 실현되여야 한다. 즉

- 사용자들은 선행획득과 그리고 애플레트원천과의 신용관계에 의거할수 없게 된다.
- 개인적으로 볼 때 매개 접근요구에서 애플레트에 의하여 이루어 진 접근요구를 따르려고 하는 사용자들은 많지 않다.
- 의뢰기의 조작체계가 어떠한 보호를 제공한다고 기대할수 없다.

이것은 Java언어설계자들자신이 체득한 환경이다. 설계자들은 조작체계환경에 의존하지 않는 애플레트를 작성하는 언어를 만들면서 모래통(sandbox)이라는 개념을 받아 들여(그림 11-4) 애플레트가 이 모래통을 떠나지 못하게 하였다. 보안에서 고려하여야 할 점들은 설계를 결정하는 여러가지 문제에 귀착된다.
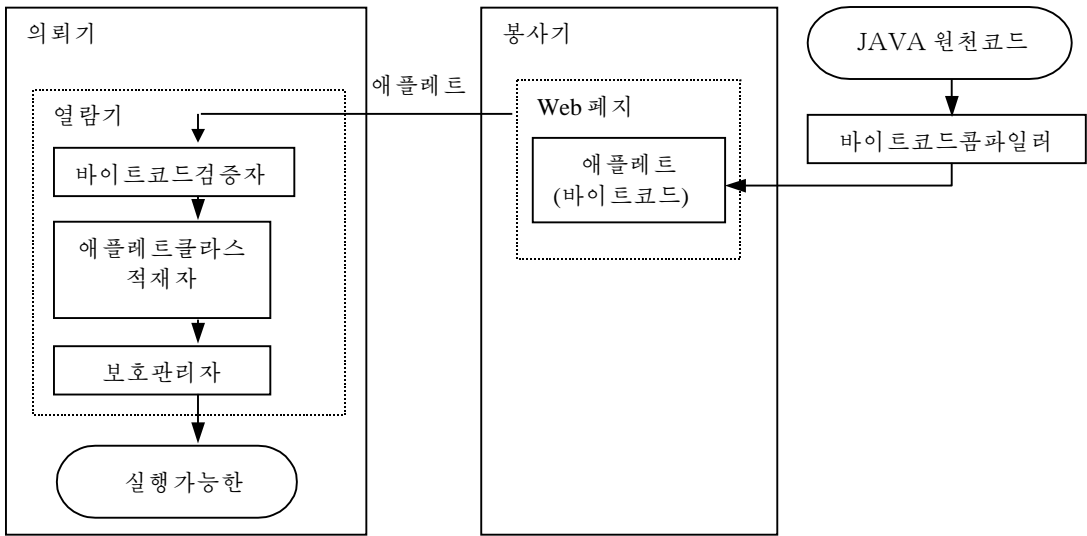
그림 11-4. Java모래통

- 언어자체는 프로그람들이 될수록 파괴를 일으키기 힘들게 되여 있어야 한다.
- 실행환경은 실행접근조종을 위한 기구를 제공한다.
- 환경에 의해서 리행되는 보안방책들은 정확히 설정되여 있어야 한다.

Java는 객체지향성이 강한 언어이다. 보안의 측면으로부터 지적자가 없으면 특별히 안전하다. 지적자에 의한 기억기접근은 C 또는 C++언어에서 오유와 보안을 파괴시킬수 있는 주되는 원인의 하나이다. Java는 형안전성을 실현한다. 객체에 기억되여 있는 클라스끄리표로 Java대상의 형을 나타낼수 있다. 정적형검사는 실행시에 취할수 있는 연산수의 인수들이 항상 정확한 형으로 되는가를 검사한다. 실행상태에서 정적형검사는 동적형검사에 비하여 복잡하지만 품이 많이 드는 작업을 사전에 하였기때문에 실행속도는 보다 빠르게 된다.

Java원천코드는 기계에 의존하지 않는 바이트코드로 변환되여 클라스파일로 기억된다. Java바이트코드는 아쎔블리언어와 비슷하다. 조작체계환경특정인 가상기계는 바이트코드를 해석하여 기계에 특정인 명령으로 변환한다. 프로그람을 실행할 때 클라스적재프로그람은 요구되는 어떤 보충적인 클라스들을 넣는다. 기성클라스들은 실행시간환경에 의하여 제공되는 《조작체계》의 또 하나의 부분을 이룬다.

Java는 일반용프로그람언어로서 독자들에게 C++대신에 Java언어로 응용프로그람을 작성할것을 권고한다. 여기서는 Java응용프로그람의 실행시 보호측면에 대해서는 론의하지 않는다. 다만 Java언어로 작성된 실행가능한 내용이면서 원격봉사기로부터 오는 보안구성방식의 Java애플레트를 실행시키려고 한다. 이런 의미에서 보안은 다음과 같은것들을 의미한다고 할수 있다.

- 애플레트들은 사용자파일체계의 접근을 가지지 않는다.
- 애플레트들은 사용자이름, 전자우편주소, 기계구성정보를 얻을수 없다.
- 애플레트들은 자기들이 온 봉사기까지만 밖으로 련결될수 있다.

- 애플레트들은 《untrusted》(믿을수 없는)라고 표시된 창문들만 꺼내기(POP UP) 할수 있다.
- 애플레트들은 새로운 클라스적재프로그람이나 새로운 보안관리기를 창조함으로써 체계를 재구성할수 없다(아래에서 고찰한다).

Java가 가능한 열람기들은 자기의 Java가상기계를 가지고 온다. Java가상기계는 바이트코드검증기(byte code verifier), 클라스적재기(class loader), 보안관리기(security manager)의 3가지 보안요소를 가진다. 현재의 열람기들은 류사한 방책들을 시행하고 있지만 이것은 반드시 그렇게 하여야 한다는것이 아니라 선택에 따른다.

1. 바이트코드검증자

바이트코드검증자(byte code verifier)는 문장론적검사를 수행하고 정적형검사를 위한 정리증명자(theorem prover)와 자료흐름해석을 리용하여 Java클라스파일을 분석한다. 검증에 의하여 다음과 같은 특성들이 담보된다.

- 클라스파일이 적당한 형식으로 되여 있다.
- 탄창들이 자리넘침이 일어 나지 않는다.
- 모든 연산수들은 정확한 형의 인수를 가진다.
- 형사이의 자료변환은 없다.
- 다른 클라스에 대한 모든 참조는 합법적이다.

바이트코드검증자는 해석프로그람의 부담을 줄인다. 그것은 코드가 정확하다는 특성이 담보되므로 실행시에 다시 검사하지 않아도 되기때문이다. 그럼에도 불구하고 보안은 여전히 실행시간환경에 의존하게 된다.

2. 애플레트클라스적재프로그람

클라스적재프로그람(class loader)는 실행시간환경의 완정성을 보호하여야 한다. 애플레트들은 자기의 클라스적재프로그람을 창조해서는 안되며 서로 간섭하지 말아야 한다. 매개 클라스적재프로그람은 자기의 이름공간을 가지고 있다. 매개 클라스는 그것을 설치한 클라스적재에 의하여 표식화된다. 애플레트들은 애플레트클라스적재프로그람에 의하여 조종된다. 망으로부터 반입된 클라스들은 자기의 출처에 근거하여 개개의 이름공간에 보관된다.

Java는 자기의 클라스서고를 가지고 온다. 이 서고에 있는 클라스들은 아주 쓸모있으며 망으로부터 오는 클라스들과 조종을 같게 해줄 필요는 없다. CLASSPATH환경변수는 기성클라스 즉 보안검사를 더하지 않아도 자동적으로 넣어 지는 클라스들이 있는 위치를 표기한다. CLASSPATH를 변화시키거나 출처가 의심스러운 CLASSPATH를 첨부할 때 보안에 미치는 영향이 뚜렷하여야 한다.

한 클라스가 다른 클라스를 참조하는 경우 애플레트클라스적재프로그람은 먼저 국부이름공간에 있는 기성클라스들을 탐색한다. 만일 요구한 클라스를 발견하지 못하였다면 탐색은 참조를 만들고 있는 클라스이름공간까지 탐색을 확장한다. 이 탐색경로를 따르면 기성클라스들을 《속여》 넘길수 없게 된다.

3. 보안관리자

보안관리자(security manager)는 Java보안모형의 참조감시기로서 《위험한》 방법들에 대하여 실행시간검사를 진행한다. 특정의 방책들을 리행하여 들어 온 클라스, 국부클라스, 기성클라스들을 구별할수 있다. Java클라스들을 다시 패키지로 구분된다. 패키지들은 클라스에로의 초보적인 접근조종을 쉽게 한다. 객체본보기에 따르면 클라스들은 변수(속성)들과 방법들을 가진다. 변수들과 방법들은 다음과 같이 선언해 줄수 있다.

- Private: 변수나 방법을 창조하는 클라스만이 접근을 가진다.
- Protected: 변수나 방법 그리고 그의 부분클라스를 창조하는 클라스만이 접근을 가진다.
- Public: 모든 클라스가 접근을 가진다.
- None of the above: 동일한 패키지안에 있는 클라스만이 접근을 가진다.

클라스들은 자기가 속한 패키지자체를 선언한다. 보안관리자는 클라스 그자체가 특권화된 패키지에 첨부되지 않도록 하여야 한다. Java의 새 판본에서는 서명된 애플레트들을 접근조종기준보관소에 첨부하였다.

4. Java보안의 현 상태

싼(Sun)회사가 고심어린 노력을 들여 보안에 달라붙었음에도 불구하고 Java보안은 지금까지 완전히 성공하지 못하였다. 그 구체적인 원인들을 [95]에서 제시하였다. 8장에서 있을수 있는 이러한 문제들을 일부 언급하였다. 대부분의 경우에 형체계를 파괴하는 방법으로 공격이 개시된다. 이것은 객체지향추상화에 강한 보안을 구축하기 위하여 시도하고 있으며 그 기초에 놓이는 객체관리체계를 확고하게 하기 위해 더는 노력하지 않아도 되기를 바라는 모든 사람들에게 알려 주는 경고로 되여야 할것이다. 복합체계를 빈틈없이 하는것은 어려운 과제이다(형체계를 파괴하는것은 《층아래의 접근을 하게 하는》 다른 하나의 실례로 된다). Java보안은 Web열람기에서의 가상기계가 하여야 할 문제이다. 다시 말하지만 보안은 조작체계우에 있는 봉사층에 놓인다. 실례로 Web열람기가 아닌 응용을 동작시킴으로써 사용자가 보안기구아래에 있는 층에로 접근하면 보안체계가 완전하다고 하던것이 수포로 되여 버리고 만다. 한편 조작체계의 보안특징들은 공개되여 있으므로 Web보안을 강화할수 있다. 끝으로 언급하고 싶은것은 Java보안모형이 하나의 틀거리를 제시하는것이지 어떤 고정된 보안방책에 맞추라고 하는것은 아니다.

제7절. 지적소유권보호

내용제공자들은 자기들의 Web폐지우에 표시되여 있는 정보로부터 소득을 얻으려고 한다. 수자정보들은 아주 쉽게 복사되여 전송될수 있다. 따라서 내용제공자들에게는 자기의 업무를 보호하는데 도움을 줄수 있는 기구가 필요하게 된다. 이것은 망보안에서 결코 새롭게 제기되는 문제가 아니다. 지난 시기에는 쏘프트웨어회사들과 음악 및 비데

오산업에서 제기되였었다. 지적소유권(IPR)의 보호에서는 지금이나 지난 시기에나 명백한 해결책이 없다. 쏘프트웨어보호를 간단히 다시 돌이켜 보면 그 리유를 알게 될것이다. 쏘프트웨어보호라고 할 때 쏘프트웨어를 허가없이 함부로 사용하지 못하게 한다는것을 말한다. 다음의 두가지 기본적인 공학적해결방도가 있다.

- **복사보호 :** 쏘프트웨어는 그것을 기억하는 하드웨어와 결합시킨다.
- **용도의 제한 :** 쏘프트웨어는 그것을 실행하는 하드웨어와 결합시킨다.

두 방법이 다 자기의 약점이 있지만 오늘날 적지 않은 쏘프트웨어판매자들은 법률제도를 통하여 자기들의 권리를 행사하고 있다고 믿고 있다. 쏘프트웨어판매자들은 콤퓨터비루스들이 굉장히 떠돌아 다니는것을 알기때문에 사용자들에게 값 눅은 복사판보다 《원본》프로그람을 사는것이 좋다고 선전하고 있다.

1. 복사보호

복사보호는 프로그람이 플로피디스크에서 보급되던 시기에 사용하였다. 디스크에 기억된 자료는 자리길과 분구로 구분된다. 분구에는 자료위치를 나타내는 머리부와 기본완정성검사를 위한 검사합이 들어 있다. 디스크에 기억된 프로그람은 표준형식과 편차날수 있으므로 판매자가 제공하는 전용루틴으로만 검색할수 있다. 복사루틴이 비규격복사를 성공적으로 진행할수만 있다면 방어대책은 없게 될것이다. 복사보호기구들의 출현은 보다 위력한 복사프로그람을 실제적으로 개발할수 있게 하였다. 다음의 선택권들이 복사보호를 위하여 연구되였다.

- **론리보호:** 《unlistable》(목록으로 표시할수 없음)기발이 설정되였거나 《invisible》(볼수 없음)문자를 만났을 때 파일이 목록으로 표시되거나 복사되지 않도록 복사와 목록루틴을 다시 작성한다. 사용자가 조작체계를 환히 꿰뚫고 있다면 보호는 불가능하다.
- **비규격화디스크형식:** 플로피디스크형식을 변화시키기 위한 선택권에는 자리길을 형식화하지 않거나 분구당 자리길수, 번호, 분구크기, 검사합을 변경하거나 라선형자리길화가 있다. 이러한 기구들은 규격디스크형식이라고 가정하는 프로그람들을 복사하지 못하도록 하는데서는 쓸모 있다. 이에 대처하여 nibble/bit(1nibble=4bit)로 복사프로그람을 작성하면 론리적형식을 무시하고 디스크의 물리적인 복사를 실현할수 있다.
- **디스크의 지문화:** 쏘프트웨어를 디스크의 고유한 물리적특성과 결합하여 원본을 판매하며 정확한 지문이 있는 디스크에서만 동작시킨다. 보통 지문으로서는 형식화된 자리길40(보통 형식화되지 않는다.)의 존재, 매개 자리길에서 0번분구를 읽는 지연시간, 자리길당 비트수 또는 고의적으로 손상시킨 디스크의 불량분구의 위치가 될수 있다.

코드가 극소형처리기나 RAM에 있는 경우에 코드에 접근하면 디스크에 의한 복사보호는 아무런 의미도 없게 된다. 실례로 새치기처리프로그람을 수정하여 어떤 프로그람이 실행될 때마다 복사수속을 호출할수 있다.《복사카드》(기억기판)를 극소형처리기의 내부모선우에 설치하고 새치기표를 변경시키면 이 카드에로 조종을 넘겨 줄수 있다.

디스크에 의한 복사보호방법의 가장 치명적인 결함은 사용자들이 《규격화된》 쏘프트웨어구성요소들을 실행시킬수 없으므로 여벌복제(back up)와 서로 다른 제품들사이의 호상조작성에 지장을 준다는것이다. 쏘프트웨어관리는 사용자측에서 벗어 나므로 사용자는 쏘프트웨어판매자들을 믿고 업무를 리용하게 된다.

2. 용도제한

복사보호에서 제기되는 기본문제는 복사의 회수보다 쏘프트웨어의 용도를 제한하면 피할수 있다. 쏘프트웨어자체가 자기가 실행하고 있는 기계의 신원을 검사할수 있다. 여기서 《신원》을 나타내는 범주에는 기계의 이름, 그의 망주소, 그의 에써네트주소 등이 속한다. 보통 이러한 값들은 어떤 기억된 값과 비교되는 검사합의 계산에 기인하고 있다. 이러한 형태의 쏘프트웨어보호는 솜씨 있는 사용자들이 다음과 같은 수단들을 받아 들이면 무효로 될수 있다.

- 오유수정프로그람. 이것은 쏘프트웨어에 의하여 검사를 분석한다.
- 쏘프트웨어의 이러저러한 조립. 이것은 쏘프트웨어가 허용되지 않는 기계에서 실행된다고 하여도 검사가 성공한다고 담보한다.

기계들을 갱신하거나 다시 이름을 붙이자면 판매자들이 검사합을 새롭게 변경하여야 하므로 비용이 증가하고 편의성은 떨어 지게 된다.

Dongle과 지능모듈은 전자장치에 대항하는 수단으로서 극소형처리기들을 소유한다면 《지능화》된다. 이 모듈은 RS-232C인쇄기대면부, 에써네트(Ethernet)대면부, 지능카드읽기장치로 콤퓨터나 극소형처리기의 내부모선에 련결된다.이 모듈은 실행시에 반드시 존재하게 되므로 쏘프트웨어는 그의 존재를 쉽게 검사할수 있다. 지능모듈은 쏘프트웨어의 극히 중요한 부분을 포함하거나 암호화된 형태로만 보관되여 있는 프로그람을 해신할수 있다. 이러저러한 프로그람의 조립품은 이 모듈과의 작용을 아주 쉽게 모의할수 있다.

사용자들은 보호된 프로그람을 Dongle에 의하여 마음대로 복사할수 있으므로 얼마든지 복사판을 만들수 있다. 동시에 실행되는 프로그람의 수만을 제한해 준다. 하나 이상의 프로그람이 보호되여야 하는 경우에는 호상조작성문제가 제기될수 있으므로 여러개의 Dongle들을 동시에 꽂아 넣어야 한다. 끝으로 사용자들은 Dongle이 고장나면 제작자들이 업무를 중단할수 있다는 사실에 대처할수 있어야 한다.

3. 지문과 내비침무늬

수자문건에 지문(fingerprint)과 내비침무늬(watermark)를 첨부하는 방법은 내용보호를 위한 해결방도로서 현재 잘 쓰이고 있는 방법이다. 개략적으로 말한다면 내비침무늬는 어떤 문건에서 지적소유권의 소유자를 식별하여야 하며 지문은 이 문건의 구입자를 식별하게 된다.

다음과 같은 여러가지 요구들이 제기되며 보통 여기에는 모순점들이 있다.

- 내비침무늬와 지문들은 문건에서 병합되기 쉬워야 한다.
- 내비침무늬와 지문들은 제거하기가 힘들거나 불가능하게 되여야 한다.
- 내비침무늬와 지문들은 화상의 질에 영향을 주지 말아야 한다.
- 내비침무늬와 지문들은 화상을 공통적으로 변경하여도 계속 남아 있어야 한다.
- 내비침무늬와 지문들은 정확한 저작권에 의하여 쉽게 검출될수 있어야 한다.

창조자가 소유권을 사용자들에게 보여 준다면 저작권침해자들이 내비침무늬를 제거할수 없도록 하는것이 가능한가? 중재자는 내비침무늬를 찾기 위한 위치를 알게 된다. 이때 저작권침해자들은 이 위치들에서의 자료들을 변경시킬수 있다. 그것은 Web열람기로부터 오는 보다 많은 안내정보를 받게 되기때문이다. 이때 조종은 대상보다도 접근조작에 초점을 두게 된다.

이 장의 문헌안내

Web보안에 대한 일반참고서로는 [128]이 적합하다고 볼수 있다. 쿠키와 인터네트 암호화는 [124]에서 취급되였다. Java 보안의 분석은 [95]에서 구체적으로 론의되였다. Web보안에 대한 매우 가치 있는 정보들은 다음의 Web위치에서 찾을수 있다.

http://www.w3.org(the homepage of the world wode wev consortium);
http://java.sun.com/sfag/index.html(JavaSoft's introduction to Java security);
http://java.sun.com/forum/security Forum.html(more on Java security);
http://hoohoo.ncsa.uiuc.edu/cgi(the de facfc CGI standard)
http://www.microsoft.com/intdev/Security/anthcode/anthwp.zip(Microsoft's authencode);
http://wwwcgi.umr.edu/~cgiwrap/intro.html(CGIWrap);

Java보안모형에서의 최근의 발전과정은 [59]에서 서술되였다. 독자들이 자기에게 있는 기계를 리용하는 경우에 어떻게 하면 되는가 하는 표상을 가지기 위하여서는 [94]를 보시오.

인터네트에서 쏘프트웨어보호(복사권, 특허법)의 법률적기초를 론의하는데서는 아직도 해결하여야 할 문제점들이 많이 남아 있다. 이러한 문제해결의 발전동향을 취급하는 좋은 참고서는 잡지 《Communications of ACM》이다. 복사보호와 복사프로그람사이의 관계를 개괄적으로 보여 주는 도서는 [62]이다. IPR보호를 위한 정보은폐화의 현재적용평가는 [4]에 보여 주었다.

련습문제

1. 자기의 Web열람기의 현재의 보안설정값들을 문서로 표기하시오. 체계에서 보안에 관계되는 정보는 어디에 기억되여 있는가?

2. Web보안에 대한 독자의 기대를 만족시키는 보안방책을 정의하고 독자의 방책을 실현하기 위한 모형을 구성하시오.

3. 동일한 열람기대화에서 전자상업응용프로그람과 콤퓨터유희프로그람이 실행되지 못하도록 보안방책과 이에 관계되는 보안모형을 형식화하시오.

4. 성능을 개선하기 위하여 열람기들은 의뢰기의 국부완충기억기에 Web페지들을 기억한다. 적의 Java애플레트가 이 특징을 어떻게 리용하면 부여 받은 권한보다 높은 준위 특권을 얻을수 있겠는가? 완충기억화가 보안의 약점으로 되는 다른 례들을 서술하시오.

5. Unix 의뢰기우에서 적의 애플레트들을 방어하기 위하여 실현할수 있는 보호방법을 서술하시오.

6. 일련의 정황에서는 실행하고 있는 체계로부터 이동코드를 보호할 필요가 있다. 도대체 어느 범위로 하면 이 목적을 달성할수 있겠는가? 실현될수 있는 보호특성과 본래 실현될수 없는 보호특성을 목록으로 나타내시오.

7. 프로그람산업은 복사권보호를 통하여 자기의 자산을 보호하려고 시도하였지만 1990년대 초에 전반에 걸쳐 이것을 포기하였다. 지금에 와서 다시 지적소유권을 위한 보호기구를 개발하기 위하여 노력하고 있다. 독자들의 견해에 의하면 현재기술의 변화가 이후의 성공을 약속할수 있겠는가?

8. 복사보호가 프로그람업무에서 계속 유익하게 될것인가? 대답할 때 문서처리기, VLSI 설계도구, 콤퓨터유희를 고려하시오.

제12장. 암호화

대체로 10년전까지만 하여도 암호화는 콤퓨터보안에 별로 큰 도움을 주지 못하였다. 당시 콤퓨터보안은 대체로 TCB, 참조감시기들, 자유 및 위임접근조종,보안모형과 체계명세의 형식적인 검증과 같은것들이였다. 이러한 견해에서 암호화는 부차적인 문제처럼 생각되였다. 통과암호를 기억하는 한방향함수들은 안전조작체계에서 리용되는 암호화기구의 명백한 실례로 된다.

오늘날에 와서 사정은 달라 져 암호화는 콤퓨터보안에서 제기되는 모든 문제를 해결하게 될 수단이라고 볼수 있다. 그러나 안전조작체계는 지난 시기의것과 마찬가지로 너무 비싸고 너무 제한적이며 사용자의 요구에 너무 멀리 떨어 져 있으므로 종식될 운명에 놓여 있다. 암호화가 과연 이 중대한 과제를 훌륭히 해결할수 있겠는가?

목적

- 완전히 다른 의도에서 암호화를 리용하는 다양한 응용측면을 설명한다.
- 암호화의 기본개념을 고찰한다.
- 암호화를 할 때 제기될수 있는 문제들의 형태와 암호화알고리듬을 리용할 때 처리하여야 할 문제들의 형태들을 리해한다.
- 암호화를 지원하는데 요구되는 콤퓨터보안특징을 설명한다.

제1절. 소 개

암호학(cryptography)은 비밀작성의 과학이다. 암호해독(cryptanalysis)은 암호를 푸는 과학이다. 암호공학(cryptology)은 암호학과 암호해독을 내포하고 있다.

현대암호학은 매우 풍부한 수학학문이다. 이 책에서는 암호학의 세부문제들을 리해하는데 필요한 수학적배경을 론의하지는 않는다. 그대신에 암호학이 콤퓨터보안에서 어떻게 리용될수 있는가를 설명하고 종종 콤퓨터보안이 암호학연구를 진행하게 하는 선결조건으로 된다는것을 지적하려고 한다.

1. 낡은 기본모형

암호화는 원래 통신보호에 기원을 두고 있다. 통신보호라고 하면 그림 12-1에서 보여 준 환경을 념두에 둘수 있다. 두개의 실체 A, B는 안전성이 담보되지 않는 통로에서 정보를 주고 받는다. 적측이 이 통로의 정보를 읽거나 지우고 삽입할수 있다면 이 통로를 마음대로 제어할수 있는 침입자로 된다. 두 실체 A와 B 는 서로 믿는다. 두 실체 A와 B 는 침입자로부터 보호되기를 바란다.
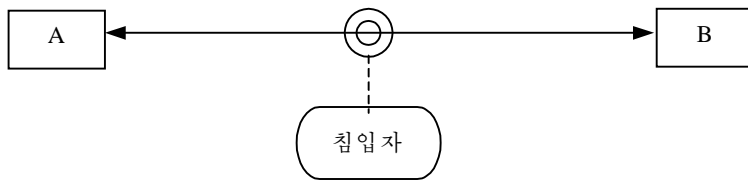
그림 12-1. 통신보안

암호학은 두 실체가 불안전한 물리적접속우에서 안전한 론리적통로를 구성할수 있게 한다. 이 점에서 암호학은 지금까지 고찰한 콤퓨터보안기구들과 기본적으로 다르다. 콤퓨터보안기구들은 모두 《아래층》(layer below)으로부터는 기능이 파괴되기 쉽다. 그러나 물리적인 통신회선에로의 접근은 암호보호를 파괴시키지 못한다.

분산체계에서는 의뢰기와 봉사기사이의 통신로가 방해를 하려고 하는 침입자들의 새로운 공격대상으로 된다. 보호되지 않는 통신회선에 의하여 생기는 약점들은 통신보안의 관점에서 볼 때 봉사와 기구들에 의하여 근본적으로 없어 질수 있다.

이러한 봉사들에는 다음과 같은 개념들이 포함된다.

- **자료기밀성:** 암호화알고리듬들은 통보문의 내용을 숨긴다.
- **자료완정성:** 완정성검사기능은 문헌이 변화되였는가를 검출하기 위한 수단으로 된다.
- **자료원본인증:** 통보문인증코드 또는 수자식서명알고리듬은 통보문의 원천과 완정성을 검증하기 위한 수단으로 된다.

여기서 말하는 자료완정성은 통신보안에 적합한 실제적인 개념은 아니다.

통신이라는 측면에서 고찰하면 통보문에는 언제나 발신자가 있다. 통보문을 받았지만 누가 보냈는지 모른다고 하면 전송도중에 변화되지 않았다고 어떻게 주장할수 있겠는가? 바로 그렇기때문에 통보문의 원천을 대조하지 않고서는 통보문의 완정성을 검증할수 없다. 한편 통보문이 전송도중에 변경된 통보문의 원천을 검증하였다고 주장하여서는 안된다. 따라서 자료원본인증이라는 범주에는 통보문완정성이 포함되며 여기서 말하는 자료완정성은 통신이 아니라 비루스제거쏘프트웨어파일을 보호하는 응용에 보다 적합하게 된다.

누가 벗이고 누가 적인가 하는 전통적인 견해는 콤퓨터보안에서 일정한 역할을 하지만 암호학을 콤퓨터작업에 응용하게 하는 주되는 힘으로는 결코 되지 못한다. 그렇지만 아직까지도 이러한 견해는 암호학에 대한 많은 사람들의 인식에서 지배적인것으로 되고 있다. 이러한 견해는 통신규약을 실현하는 여러가지 검증도구의 원리에도 반영되여 있으며 이때 A와 B가 규약의 규칙에 따라 행동하면서 침입자의 작용의 영향을 고려하기만 한다.

2. 새로운 기본모형

새로운 각도에서 고찰하자.전자상업에서 주문자는 판매자와 함께 업무거래에로 들어간다. 두 관계자들이 다 서로 속이는 일은 없다고 볼수 있지만 격렬한 흥정이 있을수 있으므로 림시방편으로 문제를 해결하는것보다도 사전에 서로 합의를 본 규칙이 있으면 틀림없이 더 좋을것이다. 따라서 주문자와 판매자는 모든 정황에서 상대방을 믿을수 있다

고 보아서는 안되는 규약을 실행해야 할 리유를 가지게 된다. 이때 반대자는 침입자가 아니라 례하면 처신을 잘못하는 어느 일방인데 이 경우 그림 12-2의 제3자가 더는 침입자로서가 아니라 신용 받는 제3자(TTP) 례하면 중재자로 된다. 비거부(non-repudiation)봉사에서는 중재자가 언제 론쟁을 해결하는가를 보여 주는 증거물을 발생한다.
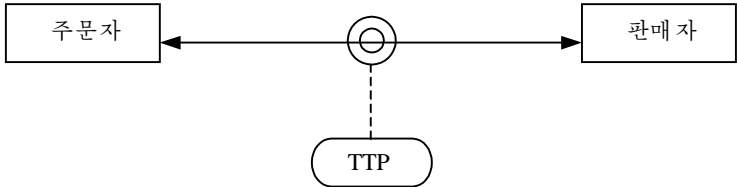
그림 12-2. 전자상업보안

적지 않은 나라들에서는 원격통신봉사제공자들이 개별적인 사용자들사이의 통신에 의무적으로 접근할수 있게 하는 도청권한을 법시행국이 언제 그리고 어떻게 줄수 있다는 것을 규정하는 법을 가지고 있다. 이제는 그림 12-3의 제3자가 합법적인 도청봉사를 제공 받아야 할 원격통신조작공의 의뢰기로 된다. 이런 정황으로 하여 통신량을 암호화하는데 리용된 열쇠를 공개시키는 열쇠날인봉사(key escrow service)가 현재 론의되고 있다.
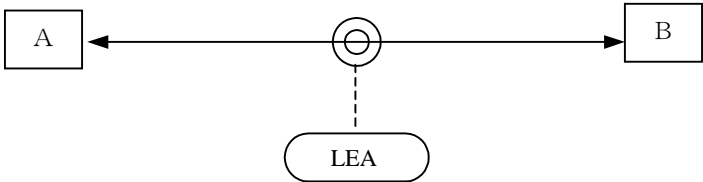
그림 12-3. 통신보안과 법시행

3. 암호열쇠

암호작성자들을 자기의 마음에 드는 아이콘들을 자물쇠로 채용하여 그들이 공개로 넘겨 주는 봉사에 신호를 보낸다. 이것은 현재의 《보안가능한》 Web열람기 또는 전자우편제품의 사용자대면부를 한번 보기만 하여도 곧 확신하게 될것이다. 류사성은 위험성을 동반하게 되므로 이것을 지나치게 분석해서는 안되지만 자물쇠제조공으로부터 암호작성자에 이르기까지 준수하여야 할 일련의 중요한 개념들이 있다. 문에 자물쇠를 채웠다가 열기 위하여 열쇠가 필요하다. 자물쇠들은 강도에서 차이가 있다. 어떤 자물쇠들은 결쇠질하기 쉽지만 어떤 자물쇠들은 너무 견고하여 침입자들은 오히려 폭력적인 공격에 의거하여 문을 뚫고 들어 가거나 완전히 전혀 다른 경로 례하면 창문으로 침입하게 된다.

암호화알고리듬은 자료를 보호하기 위하여 열쇠를 사용한다. 여기에서도 암호화강도는 다양하다. 즉 단순한 통계적방법에 의해 파괴될수 있는 방안으로부터 해석수학지식과 현재의 계산능력을 훨씬 벗어 나는 방안들이 존재한다. 폭력적인 공격은 전체 열쇠공간을 하나도 남김없이 탐색하고 알고리듬의 강도에 대한 상한을 준다.

현대암호학은 암호알고리듬의 비밀에 의거하지 않는다. 암호변환에서 쓰이는 열쇠는 보호를 하여야 할 유일한 항목으로 되여야 한다. 이 원리는 케르흐호프(Kerckhoff)에 의하여 지난 세기에 가설로 제기되였다. 이것은 서로 다른 리해관계를 가지는 사용자세계에서 준수되여야 하는 새로운 보안기본모형을 설정하는데서 특별히 적합하다. 사실상 규격화와 공개알고리듬의 열린 평가방법은 매개 동업자들에게 자기의 보안평가를 진행할 기회를 주고 새로운 당사자들이 보다 쉽게 가입할수 있게 하기때문에 이러한 정황에서는 자연스러운 과정으로 된다. 따라서 가장 일반적인 의미로서의 열쇠관리는 암호화방안들의 보안을 위한 가장 중요한 문제로 된다. 다음과 같은 물음에 답변을 주어야 한다.

- 열쇠가 어디에서 생성되는가?
- 열쇠가 어떻게 생성되는가?
- 열쇠가 어디에 보관되는가?
- 그것들이 거기서 어떻게 얻어 지는가?
- 열쇠가 실제로 어디에 쓰이는가?
- 열쇠를 어떻게 해제하고 교체하는가?

이 점에서 닫긴 고리가 이루어 지고 콤퓨터보안에로 돌아 간다. 암호열쇠는 콤퓨터체계에 기억되여 있는 자료이다. 콤퓨터체계에서 접근조종기구들은 이 열쇠들을 보호하여야 한다. 접근조종이 실패하는 경우에 암호보호는 손상을 입는다. 현재의 방법으로 분류되는 대부분의 보안체계들에서 암호알고리듬은 제일 강한 부분이므로 수가 높은 공격자들은 암호해독에 시간을 랑비하는것보다 다른 약점들을 찾아 내려고 한다.

암호학이 보안문제의 해결방도로 되는 경우는 거의 없다. 암호학은 일반적으로 통신보안문제를 열쇠관리문제로, 최종적으로는 콤퓨터보안문제로 넘기는 변환기구이다. 결과적으로 문제는 원래문제를 풀기보다 쉬워 진다. 요약하여 말한다면 암호학은 콤퓨터보안을 강화할수 있지만 콤퓨터보안을 대신하지는 못한다.

4. 모드연산

상당히 많은 현대암호화알고리듬들은 대수적원리들에 기초하여 구축되였다. 이 알고리듬들은 타원곡선이나 갈롸체(Galois)와 같은 흥미 있는 대수적구조우에서 정의될수 있다. 그러나 여기서는 조금 더 수준을 낮추어 알고리듬의 서술에서 옹근수만을 리용한다. 여기서는 모드연산의 기초로 되는 몇가지 기본적인 사실들만 표기한다.

m이 옹근수라고 하자. 이제부터는 m을 나눔수(modulus)라고 부르겠다. 이때 옹근수모임우에서 등가관계식 ≡을 다음과 같이 정의한다.

$a \equiv b \mod m$: 어떤 옹근수 λ에 대하여 $a - b = \lambda$일것이 필요충분할 때

이때 《a는 m을 나눔수로 하여 b와 등가이다.》고 말한다. ≡가 옹근수모임을 m개의 등가적인 클라스로 나누는 등가관계로 된다는것을 검사할수 있다.

$$(a)_m = \{b | a \equiv b \mod m\},\ 0 \leqq a \leqq m$$

등가클라스를 a mod m으로 표시하는것이 관례이므로 이 약속에 따른다. 다음과 같은 쓸모 있는 성질들을 확인할수 있다.

$$(a \bmod m) + (b \bmod m) \equiv (a + b) \bmod m$$
$$(a \bmod m)(b \bmod m) \equiv (ab) \bmod m$$

또한 p가 씨수일 때 모든 $a \neq 0$에 대하여 $a\ a^{-1} \equiv 1 \bmod p$가 되는 옹근수 a^{-1} 이 존재한다. 씨수나눔수 p에 대하여 p를 나눔수로 하는 곱하기차수는 다음과 같이 정의된다.

정의: p가 씨수이고 a가 임의의 옹근수라고 하자. p를 나눔수로 하는 a의 곱하기차수는 $a^n \equiv 1 \bmod p$가 되는 최소옹근수로 된다.

페르마소정리: p가 씨수일 때 모든 $a \not\equiv 0 \bmod p$ 에 대하여 $a^{p-1} \equiv 1 \bmod p$ 를 얻는다.

이 정리는 어떤 령 아닌 원소의 p를 나눔수로 하는 곱하기차수는 $p-1$의 인수로 되여야 한다는것을 주장한다. 이 사실은 암호화알고리듬의 구성에서는 거의 쓰이지 않는다. 이 알고리듬들의 보안은 보통 수론의 측면에서 다음과 같은 문제의 어려움에 관계되며 몇몇 경우에는 등가로 된다.

- **리산로그문제(DLP):** 씨수인 나눔수 p, 밑수 a, 값 $y = a^x \bmod p$가 주어 졌다면 y의 리산로그 x를 찾으시오.
- ***n*차뿌리문제:** 옹근수 m, n, a가 주어 져 있을 때 $a = b^n \bmod m$으로 되는 옹근수 b를 찾으라. 풀이 b는 m을 나눔수로 하는 a의 n차뿌리이다.
- **인수분해:** 옹근수 n이 주어 졌을 때 그의 씨수인수를 찾으시오.

파라메터들을 정확히 선택하면 이 문제들은 많은 암호화알고리듬에서 적합한 기초로 될수 있다. 그러나 이 문제들이 다같이 풀이를 구하기 힘든것은 아니다. 만일 옹근수 p 또는 n의 값이 작다면 적당한 시간내에 완전탐색에 의하여 이 문제들이 명백히 해결될수 있다. 현재 512bit옹근수이면 이미 작다고 볼수 있는 정도이며 1024bit옹근수가 일반적이라고 말할수 있다. 산수연산시간이 보다 길어 질 때 성능에서의 저하를 허용할수 있다면 물론 보다 긴 옹근수를 리용할수도 있다. 수의 길이가 고려하여야 할 유일한 론의측면으로는 되지 않는다. 이러한 난문제들은 p와 n의 구조에도 관계된다(보다 깊이 들어 가기 위해서는 이 책의 범위를 벗어 나 보다 전문적인 도서들을 읽어야 한다).

제2절. 암호화기구

암호기구들은 암호화방안의 기본구성블로크이다. 암호기구들은 암호화규약에서 쓰이며 열쇠관리가 만족된다고 보고 효과적인 보호를 제공한다. 콤퓨터보안에서 제일 많이 응용되는 암호기구들은 다음과 같은것들이다.

- 암호화알고리듬
- 수자식서명방안
- 완정성검사함수(암호화하쉬함수)

암호학의 전통적인 관례에서 벗어 나 우의 개념들을 반대순서로 소개한다.

1. 완정성검사함수

개별적응용의 요구에 따라 암호화하쉬함수를 제기하는 요구에서 미묘한 차이가 있으므로 이 절에서는 먼저 하쉬함수 h의 몇가지 기본특성들을 목록형태로 표현한다.

- 계산의 용이성: x가 주어 지면 $h(x)$를 계산하는것은 쉽다.
- 압축성: 함수 h는 비트길이가 임의로 되는 입력 x를 비트길이가 n으로 고정된 출력 $h(x)$에로 넘긴다.
- 원상저항성(한방향)(pre-image resistance(one-way)): 값 y가 주어 졌을 때 $h(x) = y$가 되는 값 x를 구하는것은 일반적으로 계산상 불가능하다.
- 2차 원상저항성(약한 충돌저항성)(2nd pre-image resistance(weak collision resistance)): 입력 x와 함수 $h(x)$가 주어 졌을 때 $h(x) = h(x')$이면서 $x \neq x'$인 다른 입력 x'을 구하는것은 계산상 불가능하다.
- 충돌저항성(강한 충돌저항성): $h(x) = h(x')$이면서 $x \neq x'$인 두 입력 x와 x'를 구하는것은 계산상 불가능하다.

조작검출코드(MDC)(변경검출코드 또는 통보문완정성코드라고도 한다.)들은 문건에로의 변화를 검출하는데 쓰이는데 다음의 두가지 특징이 있다[99].

- 한방향하쉬함수(OWHF)는 압축성, 계산의 용이성, 선상저항성, 2차선상저항성의 특성이 있다.
- 충돌저항하쉬함수(CRHF)는 압축성, 계산의 용이성, 2차선상저항성, 충돌저항성의 특성이 있다.

하쉬함수를 적용한 결과 다음과 같이 여러가지로 호출된다.

- 하쉬값
- 통보문기록집(message digest)
- 검사합(checksum)

검사합은 충분히 혼돈할수 있는 개념이다. 통신보안에서 검사합은 오유정정코드를 가리키며 대표적으로는 순환여유검사(CRC)코드를 들수 있다. 이와는 달리 항비루스제품에서 리용되는 검사합은 CRC가 아니라 암호화하쉬함수(MDC)에 의하여 계산되게 된다. x를 함부로 변경하여서는 안되는 프로그람이라고 하자. 깨끗한 환경에서 하쉬값 $h(x)$를 계산한 다음 그것을 변경시킬수 없는 장소에 즉 CD-ROM에 기억한다. 프로그람의 상태를 검사하기 위하여 하쉬값을 다시 계산하고 이것을 기억된 값과 비교한다. 하쉬값의 보호는 중요한 문제이다. 하쉬값을 계산하는데는 어떤 비밀정보도 요구되지 않으므로 누구나 다 주어 진 파일에 대하여 타당한 하쉬값을 만들수 있다.

함수 $f(x) := g^x \bmod p$는 파라메터 p와 g를 심중하게 고려하여 선택하였을 때의 한방향함수이다. 이 함수를 리산제곱이라고 부른다. 리산제곱을 역변환하기 위해서는 제12장 1절 4에서 소개한 리산로그문제를 풀어야 한다. 리산제곱함수는 후에 보게 되겠지만 암호화방식을 구성할 때 그야말로 유용한 원시함수로 된다. 그러나 리산제곱은 연산속도가 빠르다고 볼수 없으므로 방대한 량의 자료를 고속으로 처리하는 경우에는 다른 알고리듬을 찾아야 한다.

고속하쉬함수들은 흔히 류사한 설계패턴에 따라 구성하여야 한다. 하쉬함수의 핵에는 압축함수 f 가 있으며 이 함수는 길이가 고정된 입력에 작용한다. 임의의 길이를 가지는 입력 x를 주어 진 크기의 블로크 $x_1,...,x_m$ 으로 짜르며 이때 마지막블로크에 불필요한 삽입을 해준다. 그러면 x의 짜르기(하쉬)는 압축함수를 반복적으로 적용할 때 얻어 지게 된다. h_o을 (고정된)초기값이라고 하자 . i = 1,...,m 에 대하여

$$h_i = f(x_i||h_{i-1})$$

을 계산하고 그림 12-4에서 보여 준것처럼 h_m을 x의 하쉬값으로 한다(기호 ||은 련결을 표시한다).

통보문인증코드(MAC)는 통보문의 원천과 완정성(자료원본인증)에 대하여 담보한다. 두 입력, 통보문과 비밀암호열쇠로부터 MAC를 계산한다. 따라서 MAC는 때때로 열쇠하쉬함수라고 한다. MAC는 형식적으로 비밀열쇠 k에 의하여 파라메터로 표시되는 함수족 h_k이다. 함수족의 매개 성원은 압축속성과 계산의 용이성이라는 속성을 가진다. 다음과 같은 계산저항성이라는 속성이 보충적으로 성립하지 않으면 안된다.

> 상대방(적수)에게 알려 지지 않은 어떤 고정된 값 k에 대하여 $(x_i, h_k(x_i))$값들의 모임이 주어 져 있다면 어떤 새로운 입력 x에 대하여 $h_k(x)$를 계산하는 것은 계산상 불가능하다.

통보문을 인증하기 위하여 수신자는 송신자와 MAC를 계산하는데 쓰이는 비밀열쇠를 공유하여야 한다. 열쇠를 모르는 제3부류는 MAC를 유효하게 할수 없다. 다음과 같은 HMAC구성을 리용하여 MDC알고리듬 h로부터 MAC알고리듬을 유도할수 있다. 주어 진 열쇠 k와 통보문 x에 대하여

$$HMAC(x) = h(k||p_1||h(k||p_2||x))$$

을 계산한다. 여기서 p_1와 p_2은 비트렬로서 h에서 리용된 압축함수의 완전한 블로크길이에로 k를 확장한다.
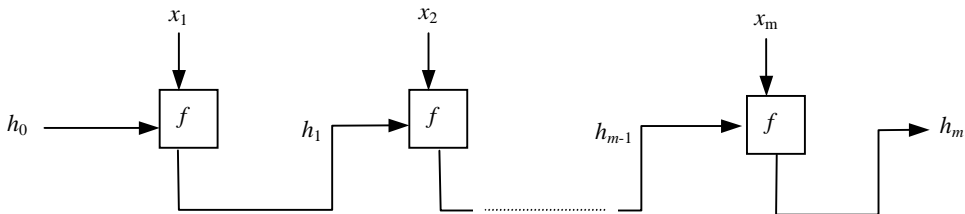
그림 12-4. 하쉬함수의 구성

안전한 하쉬알고리듬

안전한 하쉬알고리듬(SHA-1)을 택하여 실천에서 하쉬함수를 설계하는 방법을 설명하자. 이 알고리듬은 미국수자서명규격(DSA)으로 동작하도록 설계되였다. 다른 하쉬함수들은 MD4(강하지 않음), MD5(인터네트규약에서 규격선택), RIPE-MD이다. SHA-1은 512 bit블로크를 처리하고 160bit하쉬값을 발생한다. 인수들은 옹근수와 비트렬 두가지로 해석된다. 비트렬과 옹근수사이의 변환알고리듬은 이 책에서는 생략하였다.

입력은 처음에 1, 그다음에 0의 렬을 덧붙여 마지막입력블로크의 길이가 448이 되도록 늘어 나게 되며 최종적으로 64bit마당은 끼워넣기전의 입력길이를 나타내고 있다. 초기의 값은 16진수표시로 주어 진 5개의 32bit값에 의하여 결정된다. 즉

A = 67452301
B = efcdab89
C = 98badcfe
D = 10325476
E = c3d2e1f0

SHA-1의 압축함수는 80걸음으로 된 순환으로서 512 bit입력을 처리하여 20걸음마다 내부함수와 상수들을 변화시켜 160bit길이로 된 출력을 내보낸다.

내부함수는 다음과 같다.

$f_t(X,Y,Z) = (X\wedge Y)\vee((\neg X)\wedge Z)$　　　$t = 0, ..., 19$
$f_t(X,Y,Z) = X\oplus Y\oplus Z$　　　$t = 20, ..., 39$
$f_t(X,Y,Z) = (X\wedge Y)\vee(X\wedge Z)\vee(Y\wedge Z)$　　　$t = 40, ..., 59$
$f_t(X,Y,Z) = X\oplus Y\oplus Z$　　　$t = 60, ..., 79$

연산자들은 비트별 론리적, 론리합, 배타적론리합이며 32bit단어에 적용된다.상수들은 16진수형태로 다음과 같이 표시된다.

K_t = 5a827999　　　t = 0,...,19
K_t = 6ed9eba1　　　t = 20,...,39
K_t = 8f1bbcdc　　　t = 40,...,59
K_t = ca62c1d6　　　t = 60,...,79

압축함수의 시작에서 5개의 32bit변수 a,b,c,d,e들은 중간적인 하쉬값으로 초기화된다. 첫번째 입력블로크에서의 초기값 A,B,C,D,E 가 리용된다.

512bit입력블로크는 16개의 32bit단어 m_t로 부분분할되며 다음의 알고리듬에 기초하여 80개의 32bit단어 w_t 로 확장된다.

$w_t = m_t$　　　$t = 0,...,15$

$w_t = (w_{t-3} \oplus w_{t-8} \oplus w_{t-14} \oplus w_{t-16}) <<< 1$　　　t = 16,...,79

여기서 <<< s 기호는 s 비트 왼쪽으로 순환밀기를 나타낸다.

이때 압축함수는 더하기연산이 2^{32} 를 나눔수로 하여 진행되는 다음의 순환을 실행한다.

```
for t = 0 to 79 do
   begin
      temp = (a<<<5) + f_t(b,c,d) + e + w_t + K_t;
```

```
        e = d;
        c = b <<< 30;
        b = a;
        a = temp;
    end;
```

최종적으로 a, b, c, d, e 변수들에 앞단계의 중간하쉬값이 더해 지며 다음블로크를 처리할 때 초기하쉬값으로 된다.

2. 수자식서명

그림 12-1에서 통보문인증코드들은 통신통로에서 침입자가 삽입하는 부정적인 통보문을 검출하기 위한 수단으로 되므로 A와 B에게 도움을 준다. 그러나 통보문인증코드들은 제3자가 A 또는 B가 개별적으로 통보문을 보내겠는가를 결정하는데 리용할수 있는 증거로 되지 못한다. 따라서 통보문인증코드들은 주문자에게는 판매자가 주문을 날조할수 없다는 보증이 필요하며 반대로 판매자에게는 주문자가 주문의 신용을 지켜야 한다는 보증이 요구되는 그림 12-2와 같은 전자상업체계에서는 거의 쓰이지 않는다. 전자상업과 같은 경우에는 수자식서명이 필요하다.

수자식서명방안은 서명알고리듬과 검증알고리듬으로 이루어 져 있다. 문서의 수자식서명은 문서의 내용과 서명자에게만 알려 져 있는 어떤 비밀에 의존하는 값이다. 즉 비공개열쇠는 문서를 공개검증열쇠인 실체와 련관시키고 있다. 검증알고리듬은 보통 문서와 공개검증열쇠를 입력으로 선택하지만 문서 또는 문서의 일부분이 서명으로부터 회복될수 있는 경우에는 례외로 되므로 문서는 서명검증에 제공되지 말아야 한다. 다음의 검증능력의 속성은 수자식서명을 특징 짓는다.

규칙: 제3자는 서명자의 비공개열쇠를 알아야 한다는 전제가 없이도 수자식서명의 유효성에 대한 론쟁을 해결할수 있다.

수자식서명들은 비거부를 지원한다. 공개열쇠암호화(제12장 2절 3)는 수자식서명방안을 위한 자연적인 원천으로 된다. 수자식서명방안에서 비공개서명열쇠와 공개검증열쇠사이에는 검증열쇠로부터 서명열쇠를 계산상 이끌어 내지 못하도록 련결한다. 기초적인 수학적기술이 류사함에도 불구하고 수자식서명과 공개열쇠암호알고리듬사이의 계선을 명백히 그어 주어야 한다. 수자식서명과 공개열쇠암호는 기본적으로 서로 다른 목적을 추구한다. 암호화는 통보문의 기밀성을 보호하므로 가역적으로 되여야 한다. 수자식서명은 자료원본인증과 비거부를 제공한다. 수자식서명알고리듬은 가역적으로 될 필요는 없다. 사실상 가역성에 의해 보안에 관계되는 문제가 첨부된다.

1회서명

서명방안을 구성하기 위하여 그 어떤 환상적인 수학이 필요한것은 아니다. 1회서명을 얻기 위하여서는 다만 암호화하쉬함수 h[81]가 요구될뿐이다.

n-bit문서를 서명하기 위하여 우연적으로 $2n$개의 값 $x_{i,0}$, $x_{i,1}$을 선택함으로써 비공개열쇠를 따내고 $1 \leq x \leq n$에 대하여 $y_{i,0} = h(x_{i,0})$ 과 $y_{i,1} = h(x_{i,1})$을 공개열쇠로 공개한다. 이때 문서 m에 대한 서명 s의 i번째 부분은 다음과 같이 주어 진다.

$$s_i = \begin{cases} x_{i,0}\ , & m_i = 0 \text{ 일 때} \\ x_{i,1}\ , & m_i = 1 \text{ 일 때} \end{cases}$$

명백히 비공개열쇠를 다시 리용할수 없다. 이로부터 《1회서명》이라고 부른다. 검증자는 공개열쇠를 가지고 있으며 다음과 같은 검사를 한다.

$$y_{i,0} = h(s_i), \qquad m_i = 0 \text{ 일 때}$$
$$y_{i,0} = h(s_i), \qquad m_i = 1 \text{ 일 때}$$

검증자가 $y_{i,0}$, $y_{i,1}$의 값들이 진짜 공개열쇠로 되고 있는가를 확신하기 위하여 보충적인 증거물을 요구한다는것이 결함으로 되고 있는데 이것은 제12장 4절에서 보겠다.

더우기 수학적문제의 복잡성이나 어떤 다른 암호화방법의 강도에 의거할 대신 함부로 손 대기 힘든 하드웨어장치의 복잡성에 의거할수 있다. 이 장치에는 비밀서명열쇠와 비밀검증열쇠들 또는 둘중 어느 하나가 포함되여 있다. 이 장치는 검증을 위한 서명열쇠 또는 서명을 위한 검증을 사용할수 없도록 구성된다. 문서를 서명하기 위하여 이 장치는 자기의 서명열쇠를 리용하여 MAC를 구성하며 그것을 문서에 첨부한다. 이 서명을 검증하기 위하여 검증자의 장치는 서명자의 서명열쇠를 검증열쇠로 보관하였다가 이 열쇠를 사용하여 MAC를 구성하고 그것을 수신한 서명과 비교하여야 한다.

엘 가말서명과 DSA

엘 가말(E1 Gamal)서명방안은 서명하는것이 곧 비공개열쇠에 의한 암호화로 되지 않는다는것을 보여 준다[45]. p가 적당히 선택된 큰 씨수라고 하자. g가 p를 나눔수로 p–1차의 옹근수라고 하자. a 가 사용자 A의 비공개서명열쇠이고 $y_a = g^a \bmod p$가 대응하는 공개검증열쇠라고 하자.

서명되는 문서가 $0 \leqq m \leqq p$인 어떤 옹근수 m이라고 가정하자. 그렇지 않는 경우에는 적당한 하쉬함수를 적용하여 문건기록집을 서명할수 있다. m을 서명하기 위하여 사용자 A는 $\gcd(k, p-1) = 1$이 되는 우연수 k를 $0 \leqq k \leqq p$에서 선택한 다음 $r = g^k \bmod p$를 계산하고 s를 모르는 수로 하여 다음의 방정식을 푼다.

$$a \cdot r + k \cdot s \equiv m \bmod (p-1)$$

(r, s)의 쌍은 m에 대한 A의 서명을 이룬다. 검증자는 A의 검증열쇠 y_a를 요구하며 다음의 식을 검사한다.

$$y_a^r \cdot r^s \overset{?}{=} g^m \bmod p$$

서명이 정확하다면 방정식

$$y_a^r \cdot r^s = g^{(a \cdot r + k \cdot s)} = g^m \bmod p$$

가 성립한다. 이 방법에서 보안은 리산로그문제와 밀접히 관계되지만 등가적으로는 되지 않는다.

엘 가말서명방안으로부터 보다 안전하면서도 보다 효과적인 여러가지 서명방안들이 유도되였는데 대표적으로 수자식서명알고리듬(DSA)[116]을 들수 있다.

이 방안에서는 사용자 A의 비공개 및 공개열쇠들은 다음과 같이 생성된다.

1. $2^{159}<9<2^{160}$ 이 되는 씨수 q를 선택한다.
2. q가 p로 나누어 지도록 $0 \leqq t \leqq 8$인 옹근수 t와 $2^{(159+64t)} < p < 2^{(512+64t)}$인 씨수 p를 선정한다.
3. $1<\alpha<p-1$인 α를 선택하고 $g = \alpha^{(p-1)/q} \bmod p$를 계산한다.
   $g = 1$이면 새로운 α를 가지고 다시 해본다.
   (이 단계에서는 차수가 q이고 p를 나눔수로 하는 발생자 g를 계산한다.)
4. $1 \leqq a \leqq q-1$인 a를 선택한다.
5. $y = g^a \bmod p$ 를 계산한다.
6. A의 비공개열쇠는 값 a이고 공개열쇠는 (p, q, g, y)이다.

A가 문서 m에 서명을 하도록 하쉬값 $h(m)$은 SHA-1에 의하여 계산되며 옹근수로 변환한다. 그러면

7. $1 \leqq k \leqq q-1$ 인 옹근수 k를 우연적으로 선택한다.
8. $r = (g^k \bmod p) \bmod q$를 계산한다.
9. $k^{-1} \bmod q$를 계산한다.
10. $s = k^{-1}(\mathrm{h(m)}+a \cdot r) \bmod q$를 계산한다.

m에 대한 A의 서명은 (r, s)의 쌍이다. 서명은 다음과 같은 항목으로 A의 공개열쇠(p, q, g, y)에 의한 검사를 진행한다.

- $1 \leqq r \leqq q$ 이면서 $1 \leqq s \leqq q$ 인가를 검증한다.
- $w = s^{-1} \bmod q$를 계산한다.
- $u_1 = w \cdot h(m) \bmod q$와 $u_2 = r \cdot w \bmod q$를 계산한다.
- $v = (g^{u1} y^{u2} \bmod p) \bmod q$를 계산한다.
- $v = r$ 가 필요충분할 때 접수한다.

RSA서명

RSA알고리듬[127]은 발명자들인 리베스트(Rivest), 샤미르(Shamir), 애들맨(Adleman)의 이름을 딴것이며 서명과 암호에 다같이 리용될수 있다. RSA의 바로 이러한 특유한 성질은 현재 널리 류행되고 있는 수자식서명과 공개열쇠암호에 대한 여러가지 그릇된 견해를 비판하고 있는 요인으로 된다. RSA서명방안에서 사용자 A는 두개의 씨수 p와 q를 선택한 다음 $\gcd(e, p-1) = 1$, $\gcd(e, q-1) = 1$이 되는 비공개서명열쇠 e를 찾는다. 공개검증열쇠는 적 $n = p \cdot q$과

$$e \cdot d = 1 \bmod \mathrm{lcm}(p-1, q-1)$$

로 되는 지수 d로 구성된다.서명할 문서는 $1 \leqq m < n$인 옹근수 m이다. 문서에는 검증자가 문서가 진짜인가를 식별할수 있도록 여분의 정보가 충분하게 들어 있어야 한다. 원본문서가 너무 크다면 하쉬함수를 적용하고 서명에 대한 기록집을 얻기 위하여 정보를 덧붙일수 있다. m을 서명하기 위하여 A는 다음과 같이 서명을 형성한다.

$$s = m^e \bmod n$$

검증자는 A의 검증열쇠 (n, d)를 요구하고 다음과 같은 관계가 성립하는가를 검사한다.

$$s^d \stackrel{?}{=} m \bmod n$$

서명이 정확하다면 우의 방정식은 만족된다. 그것은

$$s^d = m^{(e \cdot d)} = m \bmod n$$

이 만족되기때문이다.

문서의 크기가 작으면 서명으로부터 회복할수 있으므로 따로따로 전송해서는 안된다. RSA보안은 인수분해의 복잡성에 밀접히 관계되지만 등가는 아니다.

RSA가 씨수로 되는 경우와 같은 일련의 서명방안에서는 서명검증이 특별히 빨라지도록 공개검증열쇠를 선택해 줄수 있다. 그림 12-5에서 보여 준 표는 적절히 빠른 기계에서 인수길이가 1024bit인수에 대한 DSA와 RSA의 성능과 서명하는데 요구되는 품의 개략적인 비교결과를 보여 준다.

	RSA-1024	**DSA**
초당 발생되는 열쇠수	적다	50-140
초당 서명의 회수	10-50	50-100
초당 검증의 회수	500-2000	50-100

그림 12-5. RSA와 DSA의 비교

3. 암호화

지금까지 우리는 자료의 기밀성을 보호하는 알고리듬을 위하여 암호화라는 용어를 남겨 두었다. 암호화알고리듬(암호기라고도 부른다.)은 암호화열쇠의 조종하에서 평문을 암호화한다. 해당한 복호열쇠를 가진 복호화에 의해 암호문으로부터 평문이 회복된다. 일부 암호화알고리듬들은 완정성검사를 하는 수단을 제공하지만 언제나 그렇게 되는것은 아니다. 서술된 서명알고리듬을 《비공개열쇠가 있는 암호화》라고도 생각할수 있지만 이러한 견해는 흔히 틀린것으로서 언제나 그릇된 결과를 초래한다. 이 절에서는 제10장 2절 1과 같은 표기법을 사용한다.

- $eK(X)$: 열쇠 K가 주어 진 조건에서 암호화된 평문 X를 나타낸다.
- $dK(X)$: 열쇠 K가 주어 진 조건에서 복호화된 암호문 X를 나타낸다.

암호화알고리듬에는 두가지 종류가 있다. 대칭암호화알고리듬에서는 암호화와 복호화에 쓰이는 열쇠는 같으며 이것을 비밀로 하여야 한다. 같은 열쇠를 공유하는 모든 관계자들은 서로 암호로 된 자료를 읽을수 있다. 서로 다른 관계자들이 비공개통로를 설정하기 위해서는 통로마다 새로운 열쇠가 필요하게 된다. 방대한 량의 공유된 비밀열쇠를 유지하는것은 아주 시끄러운 관리과제로 될수 있다.

비대칭암호화알고리듬(이것을 공개열쇠알고리듬이라고도 한다.)에서는 서로 다른 열쇠들이 암호화와 복호화에 쓰인다. 암호열쇠는 공개로 할수 있지만 복호열쇠는 비공개로 되여야 한다. 알고리듬의 측면에서 두개의 열쇠들이 련계가 있다는것은 명백하지만 공개열쇠로부터 비공개열쇠를 이끌어 낼수 없도록 되여 있어야 한다. 대칭암호화체계와 공개열쇠암호화체계를 구별하기 위하여 대칭체계라는 의미에서만 비밀열쇠, 비대칭체계라는 의미에서만 비공개열쇠라는 용어를 사용한다.

비밀열쇠암호화체계에서 보안관리과제는 명백히 정확한 열쇠가 정확한 장소에 들어가게 하는데 있다. 공개열쇠암호화는 관리를 훨씬 쉽게 하는것으로 생각된다. 결국 공개열쇠들은 공개되여 있으므로 보호할 필요가 없다. 이렇게 하지 않으면 공개열쇠들이 동작할수 있겠는가?

공개열쇠암호화를 리용하여 어떤 문서를 암호화하는 경우 아마도 누가 암호화된 문서를 읽을수 있는가를 알아 두어야 할것이다. 보다 일반적으로 말하여 비공개열쇠들은 당사자들을 인증하거나 또는 실례로 문서를 읽기 위하여 접근권한을 운반하는 사명을 수행한다. 이때 공개열쇠와 접근권한사이의 회선 또는 대응하는 비공개열쇠와 관계되는 당사자들사이의 회선을 담보하는것이 기본과제로 된다. 바로 이 목적을 위하여 받아 들인것이 보증서들이며 제12장 4절에서 론의한다.

암호화알고리듬들은 다시 블로크암호와 흐름암호로 구분할수 있다. 이것을 구분하는데는 다음의 두가지 기준이 있다.

- 블로크크기: 블로크암호는 보다 많은 자료블로크, 대표적으로 64bit블로크들을 복잡한 암호화함수를 리용하여 암호화한다. 블로크암호들의 보안은 암호화함수의 설계에 관계된다. 흐름암호는 보다 적은 자료블로크, 대표적으로 비트나 바이트를 실례로 비트별 배타적론리합과 같은 단순한 암호화함수를 리용하여 암호화한다. 이 구별은 모호해 져 명백한 계선이 없다. 16bit블로크를 아직도 큰 블로크로 보겠는가? 언제면 암호화알고리듬이 간단하다고 볼수 있는가?
- 열쇠흐름: 블로크암호는 동일한 열쇠밑에서 동일한 문서에 속하는 블로크들을 전부 암호화한다. 흐름암호는 끊임없이 변화되고 있는 열쇠흐름밑에서 암호화를 한다. 흐름암호의 보안은 열쇠흐름발생기의 설계에 기대를 걸고 있다. 이 정의에 따르면 귀환방식으로 된(아래에서 설명) DES는 흐름암호로 분류될수 있다.

자료암호화규격

자료암호화규격(DES)은 대칭블로크암호알고리듬중에서 고전적인 방식이라고 볼수 있다. DES는 1970년대에 미국정부규격으로 기밀에 속하는 정보를 보호할 목적으로 개발되였으며 련방정보처리규격으로 공개되였다[114]. DES는 56bit열쇠의 조종하에서 64bit평문블로크들을 암호화한다. 매개 열쇠는 기우성바이트에 의하여 64bit작업열쇠로 확장된다. 대부분의 블로크암호알고리듬에서와 마찬가지로 DES는 페이스텔(Feistel)의 원리에 기초하고 있다. 페이스텔암호기는 둥그리기연산과 같은 기본걸음을 반복한다. 둥그리기 i의 입력을 L_i와 R_i 로 절반씩 가르고 출력을 다음과 같이 계산한다.

$$L_{i+1} = R_i$$

$$R_{i+1} = L_i \oplus F(K_i, R_i)$$

여기서 F는 어떤 비선형함수이며 K_i는 그 둥그리기의 부분열쇠이다(그림 12-6).
이 연산의 역변환은 동일한 회로에 의하여 계산될수 있다. 즉

$$R_i = L_{i+1}$$

$$L_i = R_{i+1} \oplus F(K_i, L_{i+1})$$

DES에서 비선형함수 F는 32 bit입력 R_i를 48bit블로크로 확장하여 48bit부분열쇠 K_i와 비트별 배타적론리합을 계산한다. 이 중간결과를 8개의 6bit블로크로 나누고
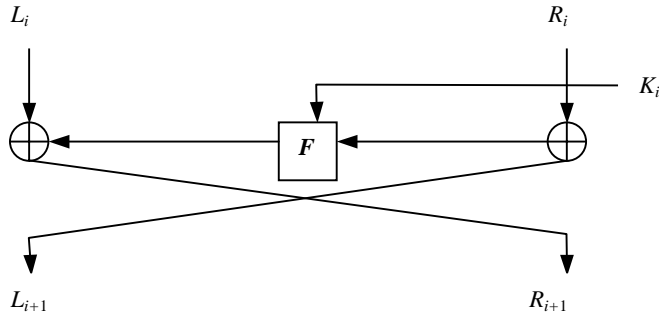
그림 12-6. 페이스텔원리

이것들을 8개의 S통(치환통)의 입력으로 리용한다. 매개 S통은 자기의 6 bit입력을 4 bit출력으로 변환한다. S통들의 출력을 P통(순렬통)으로 지나보내면서 32 bit입력에 비트별 순렬연산을 하여 F인 결과를 준다.

DES에는 이러한 둥그리기장치가 16개나 있다. 매개 둥그리기장치는 56 bit DES열쇠로부터 유도되는 서로 다른 48 bit부분열쇠 K_i를 사용하고 있다. 첫번째 둥그리기장치의 입력은 초기순렬 IP에 의하여, 마지막둥그리기의 출력은 역순렬 IP^{-1}에 의하여 처리된다. DES의 개략적인 체계를 그림 12-7에서 보여 주었다. 이 그림에서는 구체적인 열쇠순서작성알고리듬, 확장방안, S통과 순렬과정을 생략하였다.

DES가 1970년대에 규격화되였을 때 《유효수명》은 15년이라고 보았다. 그러나 DES는 아직까지도 광범히 사용되고 있으며 특히 상업 및 재정분야에서 널리 쓰이고 있다. DES보안에 대한 주되는 도전은 새로운 암호화기술이 아니라 그의 열쇠크기에로부터 오고 있다. 오늘날 56 bit열쇠공간을 완전탐색하는것은 특별한 장비가 없이도 가능하다. 워크스테이션의 성능은 해마다 계속 높아 지고 있으며 콤퓨터망의 보급으로 암호해독애호가들이 더 많은 자원들을 리용할수 있게 된다. 워크스테이션의 성능이 고정되여 있으면 56 bit열쇠는 수십년이나 몇세기가 아니라 몇주일이나 몇달정도 존재할수 있다고 기대할수 있다.

다중암호화는 알고리듬을 변화시키지 않고도 열쇠크기를 확장한다. 3개의 56 bit열쇠를 리용하는 3중DES를 선택하면 편리하다. 가장 널리 보급된 DES의 변종은 두개의 56 bit DES열쇠를 사용한 체계라고 볼수 있다. 그것은 이 체계가 하나의 DES와 《뒤방향호환성》을 가지고 있기때문이다. 이 방식에서는 두개의 56 bit DES열쇠 K_1과 K_2를 리용하여 평문 P를 다음과 같이 암호화한다.

$$C = eK_1(dK_2(eK_1(P)))$$

블로크암호방식

블로크암호들은 다양한 암호화방식에서 리용될수 있다. 전자부호책(ECB)방식에서는 동일한 열쇠밑에서 매개 평문블로크들이 독립적으로 암호화된다. 이 방식에서는 평문에 대한 정보가 루실될수도 있다. 평문블로크가 반복된다면 이것을 암호문에서 두드러 지게 나타나게 한다. 이 방식에서는 또한 완정성보호는 아주 제한되게 된다. 복호화에서는 암호문블로크의 순서가 변화되였는가, 일부 블로크들이 탈락되고 있는가, 블로크들이 이미 중복되였는가를 검출하지 않는다.
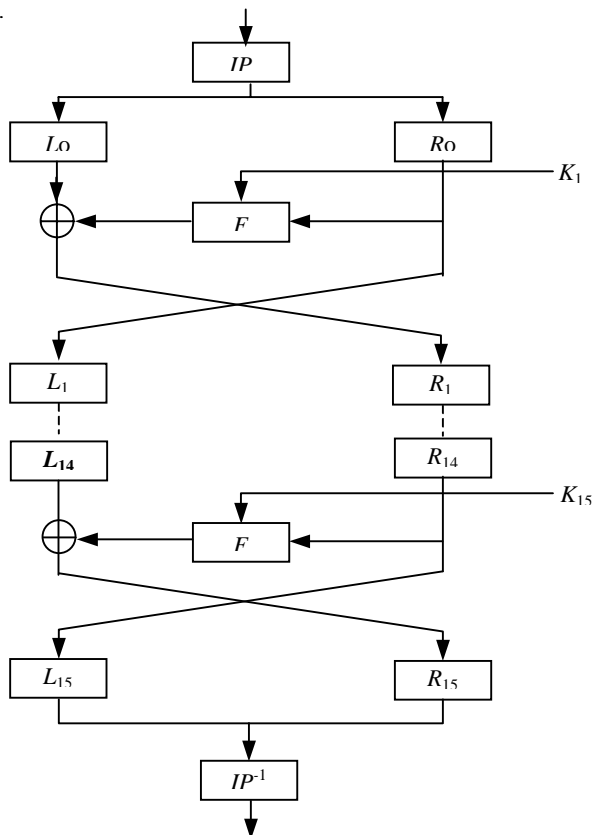
그림 12-7. DES알고리듬

암호블로크련쇄방식(CBC)에서는(그림 12-8) 앞단계의 암호문블로크 C_{i-1}이 암호화하기전에 다음평문블로크 P_i에 더해 진다(비트별 배타적론리합). 즉

$$C_i = eK(P_i \oplus C_{i-1})$$

따라서 반복된 평문블로크들은 반복된 암호문블로크로 두드러 지게 나타나지 않는다. 첫번째 평문블로크 P_1에서 초기벡토르는 C_0으로 리용된다. 비록 많은 응용들에서 초기벡토르를 비밀로 하는것은 보안의 필요조건으로 되지 않는다고 해도 일반적으로 초기벡토

르를 비밀로 하고 있다. 두개의 평문이 동일한 블로크로 시작하는것을 관측자가 검출할 수 없도록 하기 위하여 매개 통보문에 대한 초기벡토르를 변화시켜야 한다. 초기벡토르는 첫번째 암호문블로크의 복호화에서 요구된다. 암호문블로크 C_i는 다음과 같이 복호화된다.
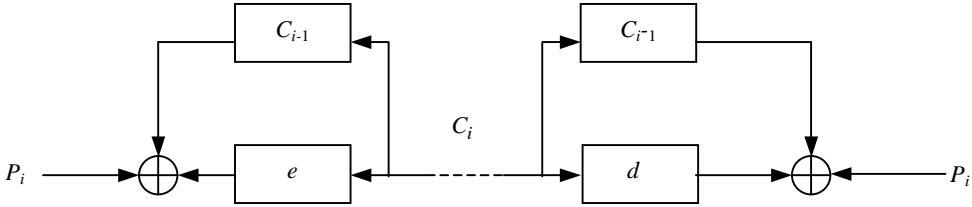
그림 12-8. 암호블로크련쇄방식

$$P_i = C_{i-1} \oplus dK(C_i)$$

만일 암호문블로크가 더럽혀 졌다면 《손상》은 다만 두개의 평문블로크로 제한된다. $\tilde{C}$이 C_i대신에 리용된다고 하자.

$$\tilde{P}_i = C_{i-1} \oplus dK(\tilde{C}_i)$$

$$P_{i+1} = \tilde{C}_i \oplus dK(C_{i+1})$$

의 조작을 한 다음 정상적인 복호화봉사를 계속한다.
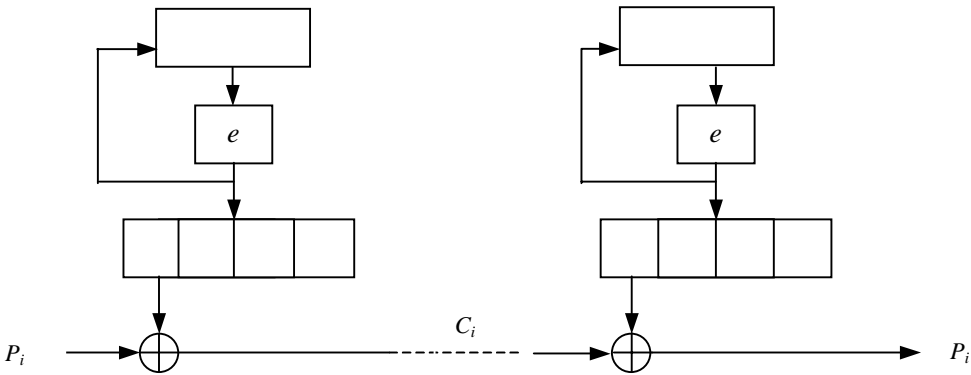
그림 12-9. 출력귀환방식

출력귀환방식(OCB)(그림 12-9)에서는 블로크암호를 흐름암호의 열쇠흐름발생기로 리용한다. 이 방식에서는 평문은 암호기알고리듬의 블로크크기보다 작은 덩이로써 처리될 수 있다. 등록기는 암호함수의 입력을 기억한다. 이 등록기의 초기내용은 초기벡토르에 의하여 결정된다. 평문덩이를 암호화하기 위하여 이것을 암호화함수의 출구로부터 오는 부분블로크에(비트별 배타적론리합) 더한다. 암호화함수의 출력은 옮김등록기에로 귀환

된다. 복호화는 암호화과정과 완전히 같다. 초기벡토르는 매 통보문에서 변화되지만 이것을 비밀로 보관할 필요는 없다. 암호문블로크 C_i의 전송과정에 발생하는 오유는 대응하는 평문블로크에만 영향을 준다. 따라서 공격자는 대응하는 위치에 있는 암호문을 변화시킴으로써 평문비트를 선택적으로 변경시킬수 있다.

암호귀환방식(CFB)(그림 12-10)에서는 블로크암호를 리용하여 자료의존형열쇠흐름을 발생한다. 이 방식에서도 평문은 암호기알고리듬의 블로크크기보다 작은 덩이로 처리할수 있다. 이 방식에서 앞단계의 암호문블로크는 옮김등록기에 귀환된다. 옮김등록기의 내용이 암호화되며 이 암호문의 부분블로크가 다음평문덩이에 더해 진다(비트별 배타적론리합). 복호과정은 암호화과정과 완전히 같다. 이 방식에서는 매 통보문마다 초기벡토르를 변화시켜 주어야 한다. 초기벡토르는 첫 암호문블로크를 복호화하는데 필요하며 비밀을 보관할 필요는 없다. 암호문블로크의 전송오유 또는 변경은 수신측의 암호화기능을 수행하는 옮김등록기에 변경된 블로크가 남아 있을 때까지 복호화에 영향을 주게 된다.
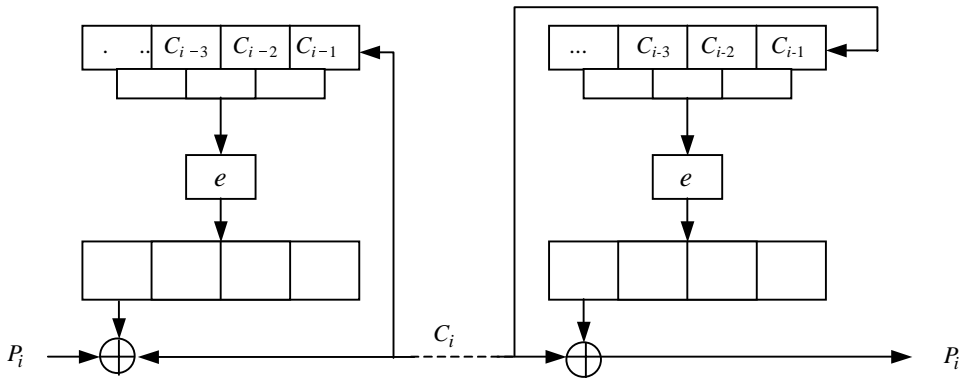
그림 12-10. 암호귀환방식

RSA암호화

설정은 RSA서명방안으로부터 잘 알수 있다. RSA가 공개열쇠암호화알고리듬으로 쓰이는 경우 사용자 A는 두개의 씨수 p와 q 그리고 $\gcd(d,p-1) = 1$, $\gcd(d,q-1) = 1$ 로 되는 비공개복호화지수 d를 고른다. 공개암호열쇠는 적 $n = p \cdot q$ 와 지수 e 로 이루어 진다. 즉

$$e \cdot d = 1 \bmod \operatorname{lcm}(p-1, q-1)$$

매개 블로크는 n보다 작은 어떤 옹근수로 되는 블로크로 통보문을 분할하여야 한다. 통보문블로크 m을 A에 전송하기 위하여 송신자는 다음식을 계산한다.

$$c = m^e \bmod n$$

이때 수신자 A는 비공개복호열쇠 d를 리용하여 다음의 결과를 얻는다.

$$c^d = m^{(ed)} = m \mod n$$

엘 가말암호화

엘 가말(E1 Gamal)공개열쇠알고리듬에서 p는 적당히 선택된 큰 씨수이며 g가 p를 나눔수로 하는 차수가 큰 옹근수라고 하자. a 가 사용자 A의 비공개복호열쇠이고 $y_a = g^a \mod p$가 대응하는 공개암호열쇠라고 하자. 매 블로크가 p보다 작은 옹근수로 되는 블로크로 통보문을 분할하여야 한다. 통보문블로크 m을 A에 전송하기 위하여 송신자는 우연수 k를 선택하고 $r = g^k \mod p$를 계산하여 다음과 같은 암호문을 A에 보낸다.

$$(c_1, c_2) = (r, m\ y_a^{\ k})$$

비공개복호열쇠 a에 의하여 A는 통보문블로크 m을 얻는다. 즉

$$c_2/c_1^{\ a} = my_a^{\ k} / r^a = mg^{ak} / g^{ak} = m$$

이 방식에서는 암호문의 블로크가 평문블로크에 비해 길이가 2배로 된다.한편 두개의 평문블로크들이 같다고 해도 대응하는 암호문블로크들은 서로 다르다. 우연수 k는 한가지 암호화에서만 사용되여야 한다. 우연수들을 다시 리용하면 암호기의 성능을 치명적으로 약화시킬수 있다.

제3절. 열쇠설정규약

암호화알고리듬을 리용하기전에 모든 열쇠들은 제자리에 설치되여 있어야 한다.

물론 이것은 우편으로 문자들을 송신하거나(실례로 신용카드에 대하여 PIN을 배포하는 일반적방법) 또는 비밀정보를 제공하는 사람을 싸이트들에 보내여 열쇠를 송달함으로써 실현될수 있다. 이러한 제안들은 안전상 측면에서나 가격상 측면에서 좋다고 말할수 없다. 열쇠관리를 현존하는 기본통신구조에 기초하여 진행하도록 하는것이 리상적이라고 볼수 있다. 다른 규약에서도 리용할수 있도록 열쇠를 설정하는 암호화규약을 열쇠설정규약이라고 부른다. 일부 열쇠설정규약들에는 공유열쇠의 설정을 바라는 관계자들만 포함시키고 있다. 그밖의 열쇠설정규약들에서는 믿음성 있는 제3자에게도 봉사를 요구하고 있다.이 두가지 경우를 구별하기 위하여 다음과 같은 규약을 론의할수 있다.

- 열쇠동의규약: 제3자의 방조가 없이도 열쇠를 설정한다.
- 열쇠전송규약: 제3자에 의하여 열쇠는 생성되고 배포된다.

암호화규약을 수학적인 세부로 분석해 보면 내부사람이 위협행위를 할수 있게 하는 《약한》 열쇠들의 부분모임이 존재한다는것을 알아 낼수 있다. 따라서 열쇠교환규약을 설계할 때에 다음의 두가지 질문에 대답을 주어야 한다.

- 약한 열쇠가 설정되여 있다면 어느 대상이 피해를 보겠는가?
- 어느 대상이 열쇠의 선정을 조종할수 있는가?

만일 그릇된 행동을 하는 내부사람이 약한 열쇠가 선정되도록 열쇠발생에 영향을 줄수 있다면 내부사람이 공격할수 있는 여지가 있을수 있다. 현재까지의 연구문헌에 의하

면 열쇠설정규약의 선정방법의 폭은 넓다. 다음에 열쇠설정규약에 기본적인 영향을 주는 두가지 대표적인 규약들을 제시한다.

1. 디피-헬만규약

디피-헬만(Diffie-Hellman)규약은 열쇠동의규약으로서[41] 비밀을 공유하지 않는 두 관계자 A와 B가 공유된 비밀열쇠를 구성하고 있다.

p가 적당히 선택된 큰 씨수이고 g가 p를 나눔수로 하는 높은 차수의 원소라고 하자.

관계자 A가 우연수 a를 선택하고 $y_a = g^a$ 를 B에 보낸다.

관계자 B는 우연수 b를 선택하고 $y_b = g^b$를 A에 보낸 다음 $y_a{}^b$를 계산한다.

$$y_a{}^b = g^{ab} = g^{ba} = y_b{}^a$$

이므로 량측이 다 비밀 g^{ab}를 공유하고 있다. 자그마한 문제가 있다. 즉 어느 관계자도 자기가 누구와 비밀을 공유하고 있는가를 알지 못하고 있다. 초기의 약속을 실현시키기 위해서는 규약에 인증을 첨부하지 않으면 안된다. [42]로부터 국사이(station-station)의 규약에 기초하여 이것을 다음과 같이 진행한다. 공유된 대화열쇠 $K:=g^{ab}$를 설정하는 디피-헬만(Diffie-Hellman)교환외에도 이 규약은 암호알고리듬과 서명알고리듬을 리용한다. 규정되여 있는 특정의 알고리듬은 없다. 아래에서 S_a와 S_b는 A와 B의 서명열쇠들이며 sS_a와 sS_b는 이 열쇠들밑에서 발생된 서명을 나타낸다. 이때 순서는 다음과 같다.

걸음 1. $A \to B$:	g^a
걸음 2. $B \to A$:	g^b, $eK(sS_b(g^b, g^a))$
걸음 3. $A \to B$:	$eK(sS_a(g^a, g^b))$

걸음 1에서 A는 우연수 a를 선정하여 디피-헬만열쇠교환을 기동한다. 그다음에 g^a를 받고 B는 우연수를 선정하고 대화열쇠 $K:=g^{ab}$를 계산한다. 걸음 2를 실행한후에 A는 디피-헬만열쇠교환의 자기의 부분을 완료하고 k를 도출한 다음 B통보문의 두번째 부분을 복호하고 B의 서명을 검증할수 있다. 마지막통보문이 도착한 다음에 B는 A의 서명을 복호하고 검증할수 있다.

2. 니드햄-슈뢰더규약

니드햄-슈뢰더(Needham—Schroeder)규약은 열쇠전송규약이다[110]. 두 관계자 A와 B는 봉사기로부터 자기의 대화열쇠를 얻는다. 초기에는 둘 다 비밀열쇠를 봉사기로써 공유한다. 암호화에 대칭암호기가 리용된다. 림시값(우연적인 도전들)들이 응답공격을 막기 위하여 통보문에 포함된다. 이 규약에서는 다음과 같은 약속기호들이 쓰인다.

K_{as}	A와 B에 의하여 공유되는 비밀열쇠
K_{bs}	B와 S가 공유하는 비밀열쇠
K_{ab}	A와 B사이의 사용을 위하여 S에 의하여 창조된 대화열쇠
N_a, N_b	A와 B에 의하여 각각 발생된 림시값

그림 12-11은 실체 A가 봉사기 S로부터 B와 통신하려는 대화열쇠 K_{ab}를 요구할 때 진행되는 걸음들을 보여 주고 있다.
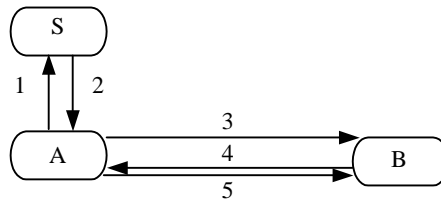
그림 12-11. Needham-Schroeder규약

걸음 1. $A \to S$: A, B, N_a
걸음 2. $S \to A$: $eK_{as}(N_a, B, K_{ab}, eK_{bs}(K_{ab}, A))$
걸음 3. $A \to B$: $eK_{bs}(K_{ab}, A)$
걸음 4. $B \to A$: $eK_{ab}(N_b)$
걸음 5. $A \to B$: $eK_{ab}(N_b-1)$

이 규약의 앞의 3걸음에서 A는 S로부터 대화열쇠를 얻고 그것을 B쪽으로 보낸다. 봉사기통보문으로 되돌려 진 림시값 N_a를 검사함으로써 A는 대화열쇠가 최근의 요구에 응답하여 발생되였으며 이전 규약의 응답공격이 아니라는것을 검증할수 있다. 마지막 두 걸음에서 B는 A가 현재 동일한 대화열쇠를 사용하고 있다는것을 검증한다.

니드햄-슈뢰더규약의 보안에 대한 세부적분석과 여러가지 보다 구체적인 보안문제들은 이 책의 범위를 벗어 나므로 여기서는 론의하지 않는다.

제4절. 보증서

국사이의 규약을 론의할 때 우리는 한가지 중요한 세부를 고찰하지 않았다.

A와 B는 자기들이 서명을 검사하기 위하여 리용하고 있는 검증열쇠가 정확하다는것을 어떻게 알것인가? 암호화열쇠와 《신원》을 련결시키는 믿음성 있는 원천이 있어야 한다. 대칭암호기에서 관계자 A와 B는 봉사기를 믿고 이러한 련계를 만들어 낸다. 공개열쇠암호기들과 서명방안들은 흔히 보증서들에 근거하고 있다.

보증권한부여(CA)는 사용자이름, 열쇠, CA의 이름, 유효기간 등을 포함하고 있는 문서를 서명함으로써 사용자와 암호열쇠사이의 련계를 보증하고 있다. 보증서의 정확한 형식을 나타내고 있는 제안들은 여러가지이지만 가장 대표적인것이라고 볼수 있는것은 X.509 Directory Framework에서 리용되는 형식이다[27]. 이 형식들은 자기들이 포함하고 있는 마당에서 일부 내용(개별적마당의 크기)에서 어느 정도 차이가 있다. 보증카드를 효과적으로 처리하는것과 광범한 응용을 위하여 유연성을 보장하는것사이에는 공학적으로 볼 때 이률배반관계가 성립하게 된다.

어떤 이름을 암호열쇠와 결합시키고 있는 보증서들은 통신보안에 기원을 두고 있다. 사용자들은 자기와 지금 말하고 있는 사람이 누구인가를 알려고 한다. 콤퓨터보안에서는 호출자의 신원보다도 호출자의 접근권한을 아는것이 일반적으로 더 중요하다. 물론 보증서를 리용하여 호출자를 인증한 다음 호출자의 사용자이름을 리용하여 접근권한을 설정할수도 있다. 그러나 가운데사람을 없애고 보증서에 접근권한을 포함시킬수 있다. 이때 대응하는 비공개열쇠는 보증서에 명기된 권한을 줄 자격을 가진다. 이 경우에 사용자의 신원을 확증하기 위하여 보증서를 검사하지 않고 오히려 자격을 인증한다.

사용자신원과 마찬가지로 보증서들은 두가지 목적에서 리용된다. 보증서들은 암호열쇠와 관계되는 실체를 식별할수 있거나(소유자의 신원을 식별하지 않고도) 암호열쇠의 소유자에게 부여되는 접근권을 나타낼수 있다.

보증서를 검증하기 위하여 CA서명을 검사하기 위한 검증열쇠가 요구된다. CA검증열쇠는 다른 CA에 의하여 담보될수 있다. 따라서 다른 보증서를 검증하기 위하여 또 다른 검증열쇠가 필요하며 이런 식으로 계속될수 있다. 바로 그렇기때문에 보증의 계층구조를 어떻게 하면 가장 합리적으로 구성하겠는가 하는 론쟁이 아직도 활기를 띠고 계속되고 있다. X.509 Directory Framework는 보편적인 뿌리보증서를 가지는 등록부나무에 보증서들을 순수한 형태로 배렬한다. 다른 극단한 경우로서 PGP(Pretty Good Privacy)보증서들은 권고체계에 기초하고 있으므로 CA를 전혀 필요로 하지 않는다. 결국 검증열쇠는 얼굴값(face value)이라고 믿어야 한다. 친구, 종업원, Web열람기로부터 또는 밖에서 또는 장부로부터 검증열쇠를 받을수도 있는데 매우 심중한 사람이라면 열쇠를 믿기에 앞서서 여러가지 원천들을 알아 볼것이다.

보증서의 취소와 관계되는 문제들은 제10장에서 이미 론의되였다.

드디여 암호보호가 암호화되지 않은 기지에 닻을 내리게 되였다.

제5절. 기구의 강도

암호화알고리듬의 강도를 평가하는것은 엄밀한 수학적토대에 기초할 때도 있고 직관성과 경험에 의거할 때도 있으므로 정확성이 없는 분야라고 말할수 있다.

암호화알고리듬은

- 경험적으로 안전하다.
- 증명가능하게 안전하다.
- 무조건적으로 안전하다.

으로 될수 있다.

알고리듬이 시간의 검사에 잘 견디면 알고리듬이 경험적으로 안전하다고 말한다. 오래동안의 분석결과에 의하면 이런 알고리듬에서 치명적인 약점이 발견된것은 없으며 알고리듬이 새로운 공격에 의해 실패하지 않을수 있다는 증명이 비록 없지만 알고리듬은 암호학계에서 인정되였다. DES는 경험적으로 안전한 알고리듬에 대한 좋은 실례이다.

차분암호해석과 같은 새로운 해석방법들은 DES의 보안을 약화시킨것이 아니라 강화하였다.

증명가능하게 안전한 알고리듬은 얼핏 보기에는 콤퓨터보안이 오래동안 바라던것(실례로 증명가능한 보안)을 제공하는것처럼 보인다. 증명가능한 보안은 복잡한 리론의 범주에서 표현된다. 만일 알고리듬을 파괴하는것이 적어도 어려운것으로 알려 진 다른 문제를 푸는것만큼 힘들다고 하면 이런 알고리듬을 안전하다고 말한다. 이 개념은 알기 힘들것 같지만 하나의 작은 단행본을 읽으면 얼마든지 리해할수 있다.

여기서 《적어도 푸는것만큼 힘들다.》고 하는것은 점근적인 개념으로서 여기에는 《아주 힘든》 실례들도 있다는것이다. 계산설비의 현재능력과 알고리듬설계의 발전을 평가하여야 한다. 례하면 우리가 인수분해할수 있는 항의 크기는 해마다 끊임없이 증가하고 있으며 독자들은 이것이 경험적인 론증으로 된다는데 동의할것이다. 지어는 더 악조건인 경우가 발생될수도 있다. 암호화에서 제기되고 있는 어려운 문제들은 인수분해하는 문제와 리산로그문제인데 반드시 풀기 어렵다는 증명은 실제적으로는 없다. 다시한번 말하지만 암호학은 지금까지 알려 져 있는 고속알고리듬은 없으며 이것을 기본적으로 돌파하기가 곤난하다는 경험적인 론의에 근거하고 있다. 보다 긍정적으로 말한다면 이 리론은 어떤 다른 암호화방식을 해독하기 힘든 정도를 론의하는데서 암호화방안을 해독하는데 요구되는 품의 아래한계를 준다는 결과를 주었다.

증명가능하게 안전한 알고리듬은 충분한 계산자원을 가진 공격자에 의하여 해독될수 있다. 물론 필요한 자원들이 어떤 공격자의 능력을 벗어 날수도 있다고 기대할수 있다. 무조건적으로 안전한 알고리듬은 무제한한 계산능력을 가진 공격자들에 의해서도 해독될수 없다.

무조건적인 보안은 정보리론으로 서술할수 있다. 공격자가 암호문의 관측으로부터 평문에 대한 보충적인 정보를 얻지 못한다면 알고리듬은 안전하다고 말한다.

무조건적으로 안전한 알고리듬의 표준적인 실례로서 한번 쓰고 버리는 암호문(1회사용암호문)을 들수 있다. 송신자와 수신자는 실제적으로 우연적인 열쇠흐름을 공유하고 한번만 리용한다.암호문은 평문과 열쇠흐름의 비트별 배타적론리합으로 된다. 수신자는 암호문에 동일한 열쇠흐름을 배타적론리합을 실시하여 평문을 회복한다.

$$\text{암호문} \oplus \text{열쇠흐름} = \text{평문} \oplus \text{열쇠흐름} \oplus \text{열쇠흐름} = \text{평문}$$

매개 열쇠가 등확률적이므로 공격자는 암호문을 보기전에는 추측할수 없는 평문에 대하여 아무것도 추측할수 없다.

지어는 무조건적으로 안전한 암호들까지도 해독되였다는것을 주의해야 한다. 조종수가 품을 줄이기 위하여 동일한 열쇠흐름을 2번 리용한다면 공격자는 두개의 암호문을 중첩시켜 놓음으로써 두 평문의 조합을 알수 있다.

$$\begin{aligned} \text{암호문1} \oplus \text{암호문2} &= \text{평문1} \oplus \text{열쇠흐름} \oplus \text{평문2} \oplus \text{열쇠흐름} \\ &= \text{평문1} \oplus \text{평문2} \end{aligned}$$

두 평문의 통보문을 중첩시켜 놓고 뜻이 통하면 아주 어렵지 않게 암호를 해독할수 있다. 베노나(Venona)계획은 이러한 사고가 발생한 일이 있다는것을 기록하고 있다.

마지막으로 극히 중요한 사실을 강조하자. 대체로 암호체계들은 알고리듬의 고유한 약점보다도 열쇠관리가 오동작하였기때문에 파괴된다. 제2차세계대전의 수수께끼는 이 점을 례증하는 가장 유명한 실례로 된다. 따라서 열쇠관리규약의 보안은 극히 중요한 문제로 된다. 보안규약의 력사는 수년동안 구체적인 여론조사를 한 다음 갑자기 새로운 공

격에로 넘어 간 규약의 력사로서 별로 깨끗치 못하다고 말할수 있다. 일부 《새로운 공격들》은 규약들을 리용하는 방법들에 대한 기본가정을 쉽게 변화시켰으며 새로운 환경에서 규약이 동작하지 않는다는것을 보여 주었다. 이 문제의 근원을 보려고 한다. 규약설계자들은 이 규약을 실현하기 위하여 지원하는 대상과 그것이 리용될수 있는 환경을 정확히 정의하기 위하여 여전히 노력하고 있다는것을 알게 될것이다.

그 유명한 실례가 실체인증방법이다. ISO작업조들까지도 실체인증이 대화열쇠의 구성을 의미하는것인지 아니면 자칭하는 신원을 검증하는데 불과한지 여러가지 견해를 세우고 있다. 이런 의미에서 《얼라이스는 보브에게 말을 한다.》 혹은 《보브가 얼라이스의 신원을 대조한다.》와 같은 의인화모형을 매우 그릇되게 해석할수 있다. 콤퓨터보안에서 실체들은 사람이 아니라 콤퓨터이고 통보문을 말로 보내는것이 아니라 망으로 보내며 A의 신원을 검증함으로써 무슨 의미인지를 결정하여야 한다. 확정적으로 말할수 있는것은 콤퓨터들사이에 시각적접촉이 없다는것이다.

얼라이스와 보브는 《달콤한 의미론적인 말》로 독자들이 그릇된 추상화준위에서 사고하도록 유혹하고 있다.

이 장의 문헌안내

비밀통신의 력사에 흥미를 가진다면 참고서 [73]을 보시오. 암호화에 대한 최근의 전문참고서로는 [99]를 리용하시오. [137]은 현대암호학에 대한 비수학적인 기초도서로 되면서 동시에 암호알고리듬의 폭 넓은 참고도서들을 제시한다. 이 책들로부터 이 장에서 제시한 알고리듬에 대한 구체적이면서도 풍부한 내용을 고찰하게 될것이다. 공개열쇠암호에 대한 시초론문은 [41]이다. 최근에 와서야 공개열쇠체계에 관한 CSEG에 의하여 초기에 분류되여 있던 연구범위로부터 벗어 나게 되였다.

http://www.cseg.gov.uk//storynse.htm

Venona문건들은 다음과 같이 찾을수 있다.

http://www.rsa.com

《진정한》 암호화규약에 관한 연구를 하는 경우에는 IP규약준위에서 열쇠동의를 위하여 Diffie-Hellman과 RSA를 채용하는 OAKLEY열쇠결정규약을 검사한다. 단순한 공개열쇠하부구조(SPKI)우에서 인터네트 Draft는 존재기간이 짧다는 특성이 있음에도 불구하고 보증리론에서 언급되여야 한다.

보증과 공개열쇠하부구조에 관해서는 다음의 주소에서 상담할수 있다.

http://www.entrust

또는

http://www.Verisign.com

련습문제

1. 암호규약은 보호되지 않는 망에서 대리인이 안전하게 통신하도록 작성된다. 이 명제가 정확한가?

2. 암호는 물리적보호를 필요로 한다. 이 명제가 어느 범위까지 정확하다고 말할수 있는가?

3. 약 n^3 개의 연산을 요구하는 n bit옹근수들에 대하여 모듈제곱알고리듬이 주어져 있는 경우에 RSA에서 512 bit로부터 1024 bit에로 이동하면 성능이 어느 정도 저하되겠는가?

4. 문건이 너무 길어서 수자서명알고리듬으로 직접적으로 처리할수 없는 경우에는 문건의 하쉬함수를 계산하고 서명한다. 공격자가 서명을 위조하는것을 막기 위해서는 이 하쉬함수에서 어떠한 특성들이 요구되는가?

- 공격자는 피해자가 서명한 통보문만을 아는 정황과 공격자는 피해자가 서명한 통보문을 선택할수 있는 정황들사이의 차이점을 구별한다.
- 공격자가 위조된 통보문의 내용을 마음대로 조종하는 선택적인 위조품과 공격자가 위조된 통보문의 내용을 마음대로 조종하지 못하는 존재적인 위조들사이의 차이점을 구별한다.
- RSA와 같은 가역적인 서명알고리듬에서 쓰이는 하쉬함수의 특정적인 요구조건을 고려한다.

5. 임의의 씨수쌍으로 RSA를 안전하게 사용할수 있는가?
   강한 씨수를 리용하는 리유와 강한 씨수로부터 일반적으로 요구되는 특성들을 따져보시오. 씨수들이 보다 길어 지면 리유가 늘 타당하게 된다고 볼수 있겠는가?

6. 엘 가말서명방안에서 우연수 k가 두개의 다른 문건들을 서명하는데 쓰이는 경우에 비공개서명열쇠를 약화시킬수 있는 방법을 이야기하시오.

7. 비공개 RSA지수를 가진 암호화는 수자서명을 창조할수 없다.이미 서명한 통보문에 검사할 여분의 정보가 없다면 공격자가 서명을 위조할수 있는 방법과 범위를 설명하시오.

8. 보증하부구조는 수자서명방안을 지원하기 위하여 필요하다. 제안범위는 X.509하부등록부나무로부터 신뢰성 있는 PGP망사이이다. 응용프로그람이 수자서명을 요구할수 있는 원인에 대해 설명하시오. 어느 보증방안이 독자가 진행한 신원확인에 가장 적합한가?

9. NP완정성문제가 암호알고리듬을 구성하는 적합한 기초로 되는가?

10. 열쇠설정규약은 내부사람이 부정행위를 쉽게 하지 못하도록 하는데만 있는것이 아니다. 열쇠교환규약을 위한 완전한 부정행위에 대한 분석을 하시오.

11. 두 관계자 A와 B가 대화열쇠발생기의 입구에 영향을 주도록 Needham-Schroder열쇠교환규약을 수정하시오.

제13장. 망보안

망들이 전개되여 콤퓨터들은 외부세계에 더많이 접근할수 있게 된다.
콤퓨터들이 외부세계를 더많이 접근할수 있도록 하면 좋은 점과 나쁜 점이 다같이 생긴다. 호상작용이 있을수록 불필요한 호상작용들도 그만큼 늘어 나게 된다. 따라서 체계우의 사용자가 어떻게 망에 접근할수 있겠는가, 망우에서 사용자가 어떻게 체계에 접근할수 있겠는가, 자료가 망에서 이동할 때 자료를 어떻게 보호하는가를 통제하려고 한다. 바로 그렇기때문에 망보안은 암호화에서뿐아니라 접근조종에도 새로운 요구를 제기하고 있다.

목적

- 망에 특유한 보안문제들을 고찰하고 망보안이 콤퓨터보안에 어떻게 이바지하고 있으며 의존하고 있는가를 리해한다.
- 기초적인 보안규약들인 IPSEC와 SSL/TLS를 실례로 하여 망보안규약의 설계에 대한 기초지식을 준다.
- 망의 경계들이 어떻게 보안의 둘레로 될수 있는가를 고찰한다.
- 다양한 방화벽들의 원리를 리해하고 이것이 제공할수 있는 봉사와 그의 고유한 제한성들을 고찰한다.

제1절. 소개

앞에서의 보안규약에 대한 론의는 얼마간 추상준위에 머물러 있었다. 자료(통보문)가 실체들사이에 전송될 때 이러한 자료교환의 정확한 본성에 대하여 전혀 관심을 돌리지 않았다. 보안규약의 일련의 특징들은 이러한 모형에서 검사될수 있지만 보안규약을 실현하는데 착수하면 콤퓨터망에 대한 구체적인 기술적특성에 보다 깊은 주의를 돌려야 한다.

콤퓨터망들은 분산체계에서 마디들사이에 자료를 전송하기 위한 통신의 하부구조이라고 볼수 있다. 하나의 마디에서 응용프로그람에 의하여 보내지는 자료는 전송을 위하여 준비되여 있다가 전자적 혹은 광학적신호렬로 전송되며 수신측에서 다시 조립되여 응용프로그람으로 제시된다. 망규약들은 송신자로부터 수신자에로의 경로를 배당하여야 하며 여기서 자료의 손실이나 이지러짐 그리고 건설자들이 전화케블을 끊었을 때와 같은 련결의 손실도 론의하여야 한다. 이것들을 총체적으로 취급하여 맨 웃층이 응용규약이고 맨 아래층이 정보비트들을 물리적으로 전송하는 규약으로 이루어 진 계층화된 방식을 리용하는것은 좋은 공학적실천으로 된다고 볼수 있다.

망보안에서 기본과제의 하나는 개별적인 보안봉사에 가장 적합한 망을 탐색하는것이다.망관리규약은 다른 통신규약에 의하여 발생되는 자료가 약속된 접수자에게 효과적으로 전달되는데 필요한 지원수단을 제공한다. 실례로 관리규약들은 송신자와 수신자사이

의 중간마디들의 리용성을 검사하고 최량적인 련결을 찾거나 또는 론리적인 망주소들을 물리적주소로 결정한다.

다른 규약들은 망마디들을 원격으로 체계에 등록하는데 리용되며 이러한 마디들에서 실행되고 있는 쏘프트웨어들은 점점 복잡해 지고 있다. 바로 그렇기때문에 망보안은 망관리규약의 보안과 망에서의 마디의 보안에 더욱더 의존하게 된다.

망마디들이 보호된 싸이트에 있다고 하여 보안이 담보된다고 하던 시기는 이미 지나갔다.

1. 계층화된 모형

ISO열린체계상호결합(OSI)방식에서는 7층모형이 망규약을 계층화하는 공통적인 틀거리로 되고 있다.이 책에서는 매층을 정확히 론의하는것보다 계층화된 모형에 관심을 가진다.그것은 계층화모형들이 망보안을 론의할 때 아주 쓸모 있는 추상화를 제공하기때문이다.

계층화모형들은 제1장 4절 2에서 고찰한 문제들을 다시 생각해 보면 쉽게 리해할수 있다.

- 상위층에서 보안봉사들은 특정의 응용에 맞게 구성할수 있다.그러나 여러가지 응용들은 매개가 다 자기의 보안규약을 요구한다.
- 하위층에서 보안봉사들은 자기보다 높은 모든 층으로부터 통신량을 보호할수 있으므로 응용규약설계자들은 보호에 주위를 돌리지 않아도 된다.그러나 일부 응용들에서는 이러한 보호가 자기들의 요구를 충분히 만족시키지 않는다는것을 알수 있다.

응용
제시
대화
전송
망
련결
물리

그림 13-1. ISO/OSI 7층모형

계층모형에서 어떤 층 N의 동위체계들은 (N)규약을 리용하여 통신한다. 층 N+1의 규약들은 N층에서 보면 가상적인 련결이라고 생각할수 있으므로 그 아래층에서 제기되는 문제들을 고려할 필요는 없다(그림 13-2).

물론 실제로 (N)규약은 보다 낮은 층들의 규약에 근거하여 구축된다.
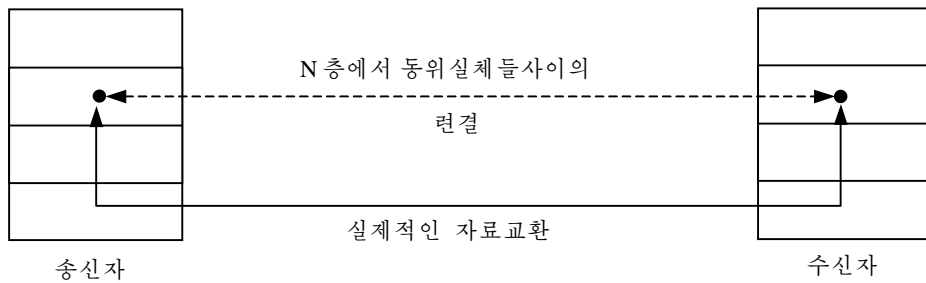
그림 13-2. N층에서 가상련결

자료를 보다 낮은 층들에 넘겨 주는 일반적인 패턴이 있다. (N)규약에서의 통보문을 (N)규약자료단위(PDU) 라고 부른다. (N)규약은 층 N-1에 있는 설비들을 불러 냄으로써 (N)-PDU를 전송한다. 이 단계에서 (N)-PDU는 단편화되여 다른 방법으로 처리될수 있으며 결과들은(N-1)-PDU로 되는 머리부와 꼬리부들이 붙는다.

(N-1)-PDU의 수신자는 머리부와 꼬리부들로부터 정보를 리용하여 (N)-PDU를 다시 조립한다. 그림 13-3은 이 과정을 간단하게 보여 준다.
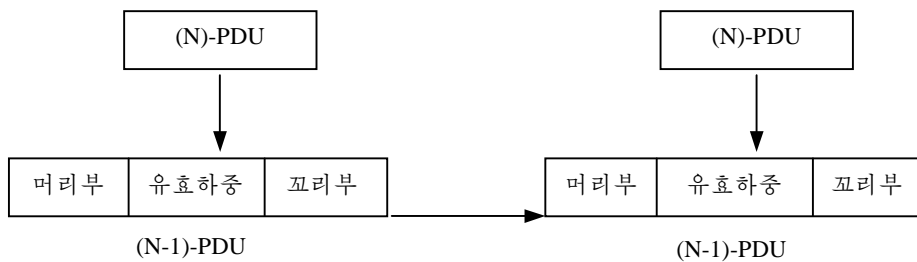
그림 13-3. (N)-PDU의 처리

이제 이 계층모형안에서 다른 봉사들에 영향을 최소로 미치는 보안봉사를 어떻게 실현할수 있는가를 간단히 보여 줄수 있다.

(N)규약이 N-1층의 보안봉사를 호출하도록 하기 위하여 현존하는 기능들중 일부를 다시 작성하거나 어떤 새로운 《보안》기능들을 첨부할수 있다.

첫번째 경우 (N)규약은 전혀 변화되지 말아야 하며 두번째 경우에는 보안기능들을 가리키도록 호출을 변화시켜야 한다. 두 경우 다 (N-1)PDU의 머리부들은 보안에 관계되는 자료를 기억하는데 편리한 위치들로 된다.

2. 탐지와 기만

통보문의 전송에 특유한 위협에 관계되는 부분은 망보안에서 한부분을 이룬다. 이러한 위협들중에서 어떤것들은 허용되지 않는 자료를 공개하거나 변경하는것과 같은것으로서 앞장들에서 충분히 론의되였다.

콤퓨터망들에서《도청》은 그다지 힘들지 않다. 송신자와 수신자사이의 직접회선은 하나의 추상화일뿐이다(그림 13-4).
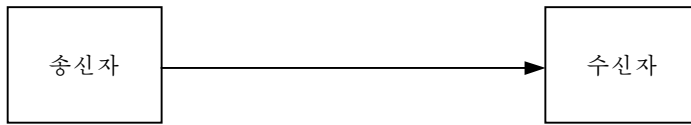
그림 13-4. 콤퓨터망에서 통신의 추상적표시

실제에 있어서 송신자와 수신자사이의 통보문을 중계하기 위하여 얼마든지 중간마디들을 둘수 있다고 생각할수 있다(그림 13-5). 이 중간마디들은 그 기능에 따라 다리, 관문, 경로기 등으로 불리운다.
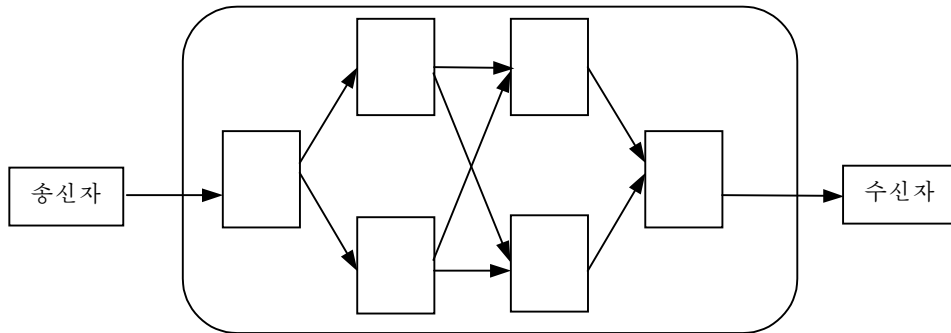
그림 13-5. 콤퓨터망에서 통보문의 전달

이러한 기구들은 들어 오는 통신량을 읽은후 다음에 처리할 동작 즉 들어 오는 파케트를 어데로 보내겠는가를 결정한다.대부분의 이러한 장치들에는 쏘프트웨어요소들이 들어 있다. 이것은 들어 오는 통신량을 읽고 민감정보를 공격자에게 넘겨 주는 사기적인 루설자쏘프트웨어가 동작할수 있는 기회를 준다.

민감정보에는 망주소 또는 실행할 준비를 갖추고 있는 마디의 규약을 지적하는것과 같은 관리자료가 들어 있다. 암호화에 의해 응용자료를 보호하는데 주의를 돌렸다고 하여도 잠재적인 공격자는 망의 내부구조를 알아 내고 민감정보를 리용하여 공격을 개시할수 있다. 마찬가지로 부하와 마디의 리용성에 대한 진단방법들을 수집한 망관리규약들로부터의 정보도 민감정보라고 볼수 있다. 동시에 이러한 규약들은 망을 효과적으로 리용하는데 필요하게 된다. 너무 지나치게 보호를 하면 망이 제공하는 봉사의 질이 떨어 질수도 있다.

이밖에도 망보안에서의 위협은 원천주소들을 위조하거나 또는 이미 참가하였던 거래에서 후에 련대책임을 거부하는 실체들로부터 또는 통신량흐름분석으로부터 생기게 되는데 여기서 공격자는 두개의 실체들이 통보문을 교환하고 있다는 사실로부터 정보를 수집할수 있다.

3. ISO/OSI보안구성방식

ISO/OSI 보안구성방식[51]은 우에서 제시한 위협들과 투쟁하는 보안봉사들을 규정한다. ISO/OSI보안구성방식은 종종 실례의 대상으로 되지만 이것을 비평하는 사람들

도 가끔 있다. 이 보안구성방식을 반대하는 론거와 이것이 아직까지 현실적으로 실현되지 못한 원인에 관한 문제는 망보안에 관한 도서에서 취급되여야 할 내용이지만 이러저러한 형태로 이것과 맞다들리는 경우가 자주 있으므로 아래에 그 봉사내용을 서술한다.

- **자료기밀성:** 통신량흐름분석으로 정보의 로출을 비롯한 권한 없는 로출을 보호하는것.
- **자료완정성:** 권한 없는 변경이나 파괴로부터 자료를 보호하는것.
- **자료원본인증:** 자료원천를 대조하는것.
- **동위실체인증:** 동위실체(동일한 규약층에 있는 실체)의 신원을 확인하는것. 흔히 동위실체인증은 련결이 이루어 졌을 때 제기되며 대화과정에 자료원본인증에 리용될수 있는 열쇠에 대한 계약이 들어 있다.
- **비거부화:** 송신자(또는 수신자)가 후에 이 사실이 틀렸다고 부인할수 없도록 자료가 이미 송신 또는 수신되였다는 증거를 만들어 내는것.
  이것은 ISO7498-2 에서 원본의 증명과 배포의 증명이라는 용어로 표현되고 있다.다른 문헌들에서는 의뢰의 증명,접수의 증명이라고 표현하기도 한다. 보통 이러한 용어의 의미들은 때때로 일치할 때가 많다.

이러한 봉사들을 제공하는데 채용된 기구들은 대부분 암호학 즉 암호화수자식서명 완정성검사함수에 의하여 실현된다. 암호학적보안은 우수한 속성을 가진다. N층에서 안전한 규약은 층아래에 있는 불안전한 규약우에서 실행될 때에도 기능이 약화되지 않을것이다.이 규칙에 한가지 례외가 있다.닉명으로 한 층에 있는 당사자의 신원을 숨기는데 주의를 돌린다면 보다 낮은 층의 규약이 첨부하는 자료는 통보문의 원천과 목적지에 대한 정보를 폭로할것이다.

제2절. TCP/IP 보안

실천에서 망보안이 어떻게 동작하는가를 구체적으로 보기 위하여 인터네트규약묶음의 범위내에서 보안을 검토한다(그림 13-6). 인터네트모형에서는 ISO/OSI 모형규약의 7개 층에서 일부는 파탄되고 4개의 층만이 남아 있다.

- 응용층에서의 규약들은 Telnet, FTP, HTTP, SMTP(단순한 우편전달규약) 또는 SET(안전전자업무)이다.
- 전송층에서의 규약은 TCP(Transmission Control Protocol)와 UDP(User Datagram Protocol) 이다. TCP와 UDP는 PDU가 포트번호에 속하는 응용층규약을 나타낸다.공통적으로 리용되는 포트번호들은 21(FTP), 23(Telnet), 25(SMTP), 80(HTTP) 이다.
- 인터네트층에는 인터네트규약(IP)가 있다.
- 대면층의 규약들은 망기술에 특정이다.

TCP와 IP는 UDP와 관리규약 ICMP와 함께 인터네트의 심장부라고 볼수 있다. TCP와 IP는 원래 믿음성이 없는 망에 련결된 사용자들에게 친절성과 협동성을 보장할 목적으로 설계되였으므로 보안문제에 전혀 관심을 돌리지 않았다. 오늘날에는 TCP/IP는 광범히 보급되고 있으므로 보안에 대한 엄격한 요구가 제기되였다. IETF(Internet

Engineering Task Force)는 인터네트와 전송층에서의 보안규약을 목적으로 제안되였다. 다음 2개 절에서 이 두가지 제안을 고찰한다. 대부분의 이 규약들은 아직까지 초안에 불과한 상태에 있으므로 여기서는 구체적인 공학적측면에로 너무 깊이 들어 가지 않는다. 이 장의 마지막에서 제시된 참고서들은 현재 기술적문헌으로 되는 원서들이다.

응용
전송/대화
인터네트
대면

그림 13-6. 인터네트층들

1. IPSEC

IP(Internet Protocol)는 인터네트층에 PDU인 IP자료본문을 전송하는 접속이 없고 국적이 없는 규약이다. IP가 접속이 없고 국적 없는 규약이므로 매개 자료본문은 다른 IP자료본문과 관계가 없는 독립적인 실체로 취급할수 있다. 또한 자료본문의 배포에 대한 담보도 없다.

IP 4판이 1981년에 RFC 791 로 공개된 때로부터 인터네트는 널리 보급되여 왔으며 IP는 새로운 요구들에 대처하는데 적응되였다.

IP 6판(IPv6)은 RFC 1883 초안으로 명세서가 작성되였고 오늘날에는 상당히 완성되였다고 볼수 있으므로 IP보안기구를 론의할 때에는 IPv6을 념두에 둔다.

IP 자료본문의 머리부는 다른 마당들에 원천IP주소뿐아니라 검사합을 포함하고 있으므로 수신자가 전송과정에 자료본문이 오염되였는가를 검사할수 있게 한다. 그러나 이 검사합은 순환여유검사에 불과하므로 자료본문을 고의적으로 변경하는 경우에는 방어할수 없다. 최악의 경우에도 수신자는 자료본문이 어디서 오는가를 실제적으로 알지 못한다. IP 머리부의 원천주소는 자료본문이 온 주소로 반드시 되는것은 아니다. 제8장 7절 1에서 제시한것처럼 원천경로공격에 의하여 이 사실을 충분히 리용할수 있다. 요약하여 말한다면 IP에는 말단-자료믿음성,자료본문의 순서화 또는 보안을 위한 특정의 기구가 없다. 이러한 보안기구들은 IP(IPSEC)를 위한 보안구성방식으로 되는 RFC 1825에서 받아 들이게 된다.

IPSEC에는 다음과 같은 두개의 기본적인 보안기구들이 포함되여 있다.

- IP인증머리부(AH), RFC 1826에 들어 있다.
- IP일봉보안 유효하중(ESP), RFC 1827 에 들어 있다.

IP보안구성방식에는 통신량분석을 막기 위한 기구들이 포함되여 있지 않다.

IP인증머리부는 IP자료본문의 완정성과 인증성을 보호하지만 기밀성을 보호하지 못한다. 그 이름으로부터 알수 있는것처럼 인증자료는 자료본문안에 있는 머리부에 놓여 있다. 그림 13-7은 IP인증머리부의 의미와 IPv6 자료본문에서 그의 위치를 보여 주고 있

다(목적지선택권들이 인증머리부앞에 있을수도 있다). 인증머리부에서 매행은 32bit단어를 나타낸다.
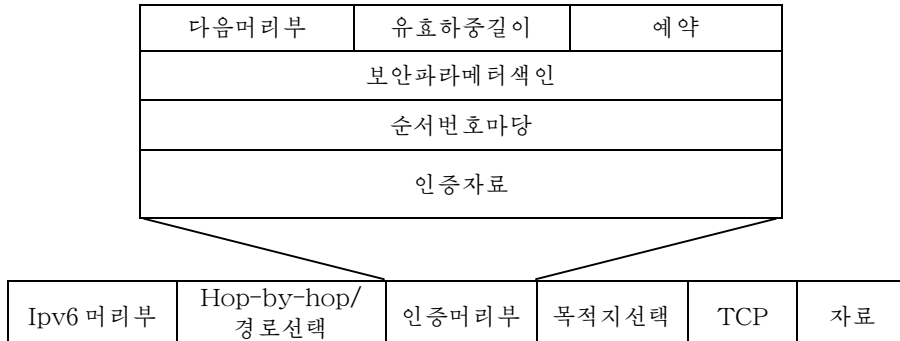
그림 13-7. 인증머리부의 위치와 의미

- **다음머리부:** 인증머리부다음에 있는 유효하중의 형태를 식별하는 8bit마당
- **유효하중길이:** 32bit단어에 있는 인증머리부의 길이-2로 나타내는 8bit마당. 실례로 인증값이 96bit일 때 인증머리부의 실제적인 길이는 6이지만 유효하중의 길이는 4로 주어 진다. IPv6에서는 값 2는 인증알고리듬이 없다는것을 나타낸다.
- **예약:** 앞으로의 사용을 위하여 예약되여 있는 16bit마당
- **순서번호마당:** 계수기의 값이 들어 있다. 이 값은 송신자에 의하여 포함되지만 수신자의 마음대로 처리된다.
- **보안파라메터색인(SPI):** 자료본문의 보안련관을 식별하는 32bit마당. 값 0은 보안련관이 없다는것을 나타낸다.
- **인증자료:** 인증자료 즉 MAC나 수자식서명과 같은 인증자료를 포함하고 있는 32bit 단어의 변수번호

자료본문을 인증하기 위하여 송신자는 처음에 완정성검사알고리듬, 암호열쇠,인증자료의 크기와 같은 파라메터를 제시하고 있는 보안련관의 위치를 지적한다.

일반적으로 사용자신원,목적지주소, SPI는 어느 보안련관을 리용할것인가를 결정한다. 보통 MAC알고리듬이 인증에 리용된다. HMAC에서 MD5 와 SHA1을 선택해 주면 암시적으로 모든 IPSEC실현에서 지원하여야 할 기정선택이다.

그러나 기타 완정성검사함수들도 리용할수 있다. 인증자료를 계산할 때 자료본문의 마당들이 수신측에서 나타난다고 본다.

IPv6 머리부에 있는 hop한계와 같은 일부의 마당들은 전송도중에 변화된다. 인증머리부에 있는 인증자료와 같은 일부 마당들은 아직 알지 못하며 MAC를 계산할 때 이 마당들에 령이 들어 간다.그다음 인증머리부에 있는 인증자료마당에 MAC가 삽입된다.

자료본문의 수신자는 해당한 보안련관의 위치를 나타내는 SPI와 목적지주소를 지적하며 인증자료를 검증한다.인증이 실패하면 실패하였다는 정보가 체계에 등록되여야 하며 자료본문은 버리게 된다.

이 알고리듬에서 IP머리부의 일부 마당들은 보안기구에 포함되지 않는다. 보호를 한층 더 강화하기 위하여 턴넬방식에서는 대표적으로 보안관문의 주소와 같은 어떤 다른 주소를 포함하는 외부IP머리부를 첨부한다. 내부IP머리부에는 초기원천과 목적지주소들이 들어 있으며 인증머리부에 의하여 완전히 보호된다(그림 13-8).

새 IP머리부	제시되는 경우에는 외부머리부	인증머리	원래 IP머리부	제시되는 경우에는 외부머리부	TCP	자료

그림 13-8. 턴넬방식에서 인증머리부

IP교갑화의 유효하중들은 기밀성을 보호하며 사용되는 암호화알고리듬에 따라 완정성과 인증성도 보호할수 있다.

ESP 머리부가 흔히 암호화된 자료의 앞에 놓인다(그림 13-9).
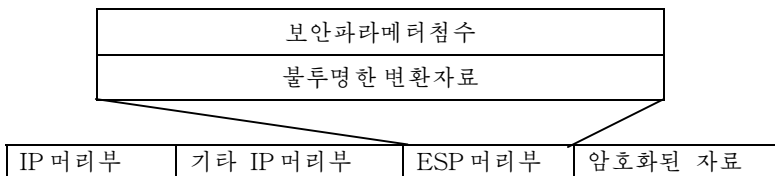
그림 13-9. ESP 머리부의 구조와 의미

ESP머리부에는 SPI가 들어 있다. 불투명한 변환자료는 암호알고리듬의 처리에 관계되는 그밖의 파라메터들을 포함하고 있는 보호마당이다.

자료본문을 암호화하기전에 송신자는 어떤 암호알고리듬과 열쇠를 사용하겠는가를 결정하기 위하여 봉사련합의 위치를 다시 지적한다(이 보안련관은 인증머리부에서 쓰이는 것과 다르다).

그러면 송신자는 다음의 두개 ESP방식중에서 선택한다.

- 전송방식에서는 TCP 또는 UDP로부터 오는 웃층프레임을 ESP에서 교갑화한다. IP머리부는 암호화되지 않는다. 전송방식은 두 마디사이에 교환된 자료본문의 말단사이의 보호를 실현한다.
- 턴넬방식에서는 완전한 IP자료본문을 ESP안에서 밀봉한다. 이 ESP는 다른 IP자료본문안에서 평문머리부에 의하여 전송된다. 따라서 IP를 턴넬화하면 IP안에서 IP를 서술할수 있다. 턴넬방식은 관문기계(방화벽)사이에서 리용되므로 가상사설망(VPN)(제13장 3절)을 창조할수 있다.

자료본문의 수신자는 해당한 보안련관의 주소를 지적하여 암호화된 유효하중을 복호화한다. 복호화가 실패하는 경우에는 실패하면 그 고장은 체계에 등록되여야 하며 자료본문을 버리게 된다.

지금까지 IPSEC범위안에서 열쇠관리문제점들에 대하여 설명하였고 열쇠현상은 그것을 요구할 때마다 제자리에 있었다. IPSEC는 보안련관과 대화열쇠를 설정하는 열쇠관리규약에 무관계하게 인증과 암호화봉사를 표현하고 있다. 따라서 IPSEC보안봉사들은 어

떤 특수한 열쇠관리규약에 얽매여 있지 않다. 열쇠관리규약에서 결함이 발견되였다고 하여도 IPSEC의 실현에는 영향을 미치지 않게 하면서도 이 규약을 교체할수 있다.

요약

IPSEC는 IP에 대한 대면부를 변화시키지 않고도 누구나 IP를 리용할수 있게 하는 보안을 제공한다(그림 13-10).상위층규약들은 보안을 불러 내기 위하여 변환할 필요가 없으며 상위층의 통신량이 IP준위에서 보호되여 있다는것을 알고 있을 필요조차도 없다. 그러나 보안준위를 응용의 요구에 맞게 맞출수 있는 여유는 그리 많지 않다.

그림 13-10. IP보안

IP는 그의 성능을 통신규약과 관련시키므로 보안련관을 선택하기 위하여 응용에 맞는 자료를 검사하는데 지나친 시간을 소비할 필요는 없다.

IPSEC에서는 송신자와 수신자가 암호화연산을 할 때 규약처리비용과 통신지연시간이 증가된다.IPSEC는 모든 웃층규약들에 대한 보안을 제공할수도 있지만 이 경우 총체적으로 볼 때 부차적인 처리들도 생겨 나게 된다. IPSEC는 특정한 열쇠관리규약을 서술하지 않으므로 서로 다른 마디들이 자기의 요구에 맞는 방안을 선택할수 있지만 방안에 동의한 다음에야 IPSEC를 사용하여 서로 다른 마디들사이 통신량을 보호할수 있다.

2. SSL/TLS

TCP규약은 두 마디들사이에 신뢰성 있는 바이트흐름을 제공한다. TCP는 국적 있는 접속지향규약으로서 파케트가 루실된 시간과 파케트가 순서대로 도착하지 않은 시간을 검출하며 반복된 자료는 버린다. TCP는 두 마디들사이의 대화를 설정할 때 지어는 주소에 기초한 신원인증까지도 진행되지만 제8장 7절 1에서 강조한바와 같이 이 규약의 실천에서는 불비한 점들이 있다. TCP는 강한 암호화실체인증과 자료완정성,기밀성이 부족하다.이러한 봉사들은 네트스케이프(Netscap)에서 주로 WWW통신량을 보호하기 위하여 개발한 SSL(Secure Socket Layer)규약에 도입되였다.전송층보안(TLS)우에서의 IETF초안은 전반에 걸쳐 SSL 3판(SSLv3)과 같으며 현재는 SSL/TLS로 알려 지고 있다.

인터네트규약안에서 보면 SSL은 응용층과 TCP사이에 위치하고 있다. 이것으로 하여 SSL은 TCP가 담보하는 속성을 믿을수 있으므로 례하면 자료가 믿음성 있게 배포되였는가 하는데 관심을 돌릴 필요는 없다. TCP와 마찬가지로 SSL은 국적이 있으며 접속지향적이다. SSL대화상태에는 암호화알고리듬의 실행에 요구되는 정보들이 들어 있는데 여기에는 대화식별자, 암호묶음의 명세서, 공유된 비밀열쇠, 보증서, 디피-헬만(Diffie-

Hellman) (제12장 3절 1)과 같은 규약에서 쓰이는 우연값과 같은것들이다. 열쇠관리에 의하여 생기는 간접처리를 포함시키기 위하여 하나의 SSL에 여러개의 접속을 포함시킬수 있다. 의뢰기와 봉사기사이의 HTTP대화가 특징적인 실례라고 볼수 있는데 이때 구성문건의 매 부분을 전송하기 위하여 새로운 접속이 이루어 진다.

그림 13-11. SSL층

상태정보의 부분모임만을 매개 접속에서 변화시켜야 한다. SSL에는 2개의 성분이 있다(그림 13-11).

- SSL기록층
- SSL응답확인층

SSL기록층은 상위층규약으로부터 블로크를 선택하여 SSL평문기록으로 단편화한 다음 현재의 대화상태에서 암호명세에 의하여 정의되는 암호학적변환에 적용한다. SSL기록층은 본질에 있어서 IPSEC에 류사한 봉사를 제공하므로 IPSEC보안련관들과 SSL상태를 대비하는것은 결코 별개의 문제로 되지 않는다.

SSL응답확인(handshake)규약은 대화상태가 있는 암호화파라메터를 결정한다. 그림 13-12는 의뢰기와 봉사기사이에 교환된 통보문을 보여 주고 있다. 괄호안에 있는 요소들은 선택성을 가진다. 이 규약을 실례로 보여 주기 위하여 의뢰기가 봉사기를 인증하는 동작을 단계별로 고찰하자.

의뢰기 ClientHello는 통보문으로써 규약실행을 시동한다. 이때 Client Hello통보문에는 우연수, 의뢰기가 선택에 따라 순서화된, 제안된 암호의 목록, 제안된 압축알고리듬이 들어 있다. 즉

```
M1: Client Hello: Client Random[28]
                  Suggected Cipher Suites:
                    TLS_RSA_WITH_IDEA_CBC_SHA
                    TLS_RSA_WITH_DES_CBC_SHA
                    TLS_RSA_EXPORT_WITH_DES40_CBC_ SHA
                  Sugested Comeression Algorithm:NONE
```

봉사기는 제안된 묶음에서 암호 TLS_RSA_WITH_DES_CBC_SHA를 선택한다.

RSA는 열쇠교환을 위하여 DES_CBC는 암호알고리듬으로, SHA는 하쉬함수로 리용한다. 봉사기는 Server_Hello통보, 보증서사슬로 응답한다.
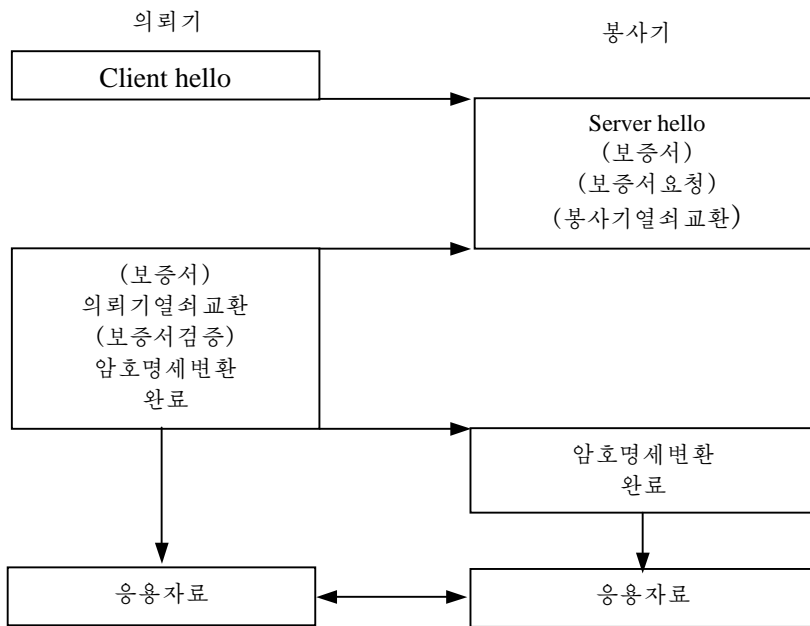
그림 13-12. SSL 응답확인규약

M2: ServerHello: ServerRandom[28]
Use Cipher Suite:
TLS_RSA_WITH_DES_CBC_SHA
Session ID:OXA00372d4xs
Certificate: Subject: DN=SuperStoreVirtualOutlet
PublicKey: Ox521aa593...
Issuer: SuperStore HQ
Subject: DN=SuperStoreHQ
PublicKey: Ox9f400682...
Issuer: Verisign
Server Done: NONE

이 실례에서 의뢰기로부터 요구되는 보증서는 없다. 의뢰기는 보증서사슬을 검증하고 우연수 48바이트 PreMasterSecret를 국부적으로 창조한다.

Master Secret의 처음 48바이트는 다음과 같다.

PRF(PreMasterSecret, 'master secret', ClientRandom || ServerRandom)

여기서 PRF는 MD5와 SHA에 기초하여 더 복잡한 함수를 간략한것이다. Mastersecret는 다음과 같은 형태로 표시되는 열쇠블로크를 구성하는 입력으로 된다.

PRF(MasterSecret, 'Key expansion', ClientRandom || ServerRandom)

기호 ||은 련결을 의미한다. 의뢰기와 봉사기에 대하여 모든 요구된 MAC와 암호화열쇠들은 열쇠블로크에서 추출한다. 의뢰기로부터 봉사기에로의 통신량을 보호하는 열쇠는 봉사기로부터 의뢰기에로의 통신량을 보호하는 열쇠와 다르다. 따라서 관계자들은 자기가 보낸 통보문과 자기가 받은 통보문을 구별할수 있으므로 통보문을 송신자에게 다시 보내는 반사공격을 받지 않게 된다.

의뢰기는 선택된 암호묶음에서 지적된 열쇠관리알고리듬과 봉사기가 보증하는 공개열쇠를 리용하여 PreMasterSecret를 봉사기에 전송한다(그다음 의뢰기는 PreMasterSecret를 즉시에 파괴시켜야 한다). 이 례에서 알고리듬은 RSA이며 공개열쇠는 Ox521aa593이다. 의뢰기는 변환을 요구하지 않고 선택된 암호묶음을 접수한다는것을 알려 주면서 세번째 통보문을 MD5와 SHA에 의하여 구성된 두개의 하쉬함수들을 통하여 앞의 두 통보문과 련결시킨다.

M3: A: ClientKeyExchange: SA_Encrypt(ServerPublicKey, PreMasterSecret)
B: ChangeCipherSpec: NONE
C: Finished MD5(M1||M2||M3A)
SHA(M1||M2||M3A)

봉사기는 PremasterSecret를 복호하고 그로부터 MasterSecret와 열쇠블로크 그리고 의뢰기와의 이 대화에 유효한 모든 비밀열쇠들을 계산한다. 봉사기는 의뢰기통보문에 첨부되여 있는 하쉬함수를 확인하고 어떤 변환을 요구함이 없이 선택된 암호묶음을 접수한다는것을 알려 주면서 다음과 같이 응답한다.

M4: A: ChangeCipher Spec: NONE
B: Finished MD5(M1 || M2 || M3A || M3C)
SHA(M1 || M2 || M3A || M3c)

의뢰기는 봉사기의 통보문에 있는 하쉬함수를 확인한다.

이때 두 관계자들은 공유비밀열쇠를 설정하였으므로 응용통신량을 보호할수 있다.

요약

오늘날 SSL은 매우 광범히 리용되고 있는 인터네트보안규약으로서 모든 Web열람기들에서 지원되고 있다. SSL은 응용규약과 TCP사이에 보안층을 첨부하였으므로 응용들은 보안을 명시적으로 요구하여야 한다. 따라서 응용코드는 변화되여야 하지만 요구되는 변화들은 SSL의 이전 응용에서 TCP connect호출을 SSL-connect호출로 바꾸는 편집조작보다 많치 않다. SSL-connect호출은 암호상태파라메터들을 초기화하여 원래의 TCP connect호출을 한다 .

SSL명세서는 IPSEC와 달리 신호교환규약을 정의함으로써 의뢰기와 봉사기가 암호묶음에 동의하여 필수적이면서도 관건적인 자료들을 설정하고 서로 인증을 한다. 또한 SSL은 IPSEC와 달리 모든 실현에 반드시 포함되여야 하는 SSL 규격에 만족되는 암호알고리듬의 묶음을 아직까지 규정하지 못하고 있다.이 점은 SSL/TLS를 인터네트규격으로 되게 하는데서 현재 장애물로 되고 있다.

제10장 3절에서는 보안문맹자의 응용을 어떻게 보호하겠는가 하는 문제점들을 제기하였다. GSS-API와 비슷한 정황이 IPSEC와 SSL에서도 제기된다. 암호봉사호출을 불러 내여 자료의 암호화 또는 자료의 인증을 진행하기 위해서 응용작성자는 응용코드를 조금(SSL) 또는 전혀(IPSEC) 바꾸지 않아도 된다.《보안문맥》은 SSL에 의하여 설정되며 IPSEC에서는 따로따로 설정되여야 한다. 의뢰기와 봉사기는 자기 봉사문맥의 파라메터를 보호하여야 하며 그렇게 하지 않은 경우에는 IPSEC에 SSL에 의하여 제공되는 보안은 약화되게 될것이며 다시 콤퓨터보안에 되돌아 가게 된다.

통신회선우에서 층아래로부터는 암호보호의 기능을 약화시킬수 없다. 암호보호의 기능은 망마디에서 층아래로부터 심히 약화될수 있다.

제3절. 망경계

IPSEC와 SSL에서 보안둘레들은 콤퓨터망의 마디들의 경계들과 일치한다.마디들은 안전하지만 망은 안전하지 못하다고 가정한다.이것은 세계에 대한 견해이다. 망들은 겹싼 부분망으로 이루어 져 있고 이러한 부분망의 경계는 적합한 보안둘레로 된다는것이 아마 사실일것이다.

국부망(LAN)을 설치한 어떤 조직을 실례로 고찰하자. 파케트들이 LAN의 임의의 마디를 경유할수 있는데 매개 마디가 다른 마디로 가게 되여 있는 파케트들을 루설할수 있는 잠재적가능성이 있다. 기관은 자기의 구내에서 도청공격이 정보를 로출시킬수 있는 다른 방법들에 비하여 기본위협은 아니지만 종업원자료가 로동행정부밖으로 우연히 나가지 않도록 할것을 결정하였다. 이를 위하여 LAN을 하나의 경로기(그림 13-13)에 련결된 두개의 부분망으로 가른다. 로동행정과에 있는 기계들은 subnet_1에 련결되여 있으며 기관의 모든 다른 기계들은 subnet_2에 련결되여 있다. 경로기는 파케트들이 다른쪽에 있는 마디가 명시적으로 주소화되여 있는 경우에만 부분망사이에서 파케트를 통과시킨다.

두개의 부분망이 직접적으로 련결되여 있지 않는 경우에는 매개 부분망에 있는 관문들사이에 안전한 련결을 설정해 줌으로써 가상사설망(VPN)을 창조할수 있다(그림 13-14). 부분망들사이의 모든 통신량은 보안둘레를 확장하기 위하여 암호학적보안이 첨부되는 이 관문을 통과하지 않으면 안된다. 우의 두 례에서 망경계의 마디들이 보안기구를 실현한다는것을 알수 있다. 일반적으로 망의 경계에 있는 기계들은
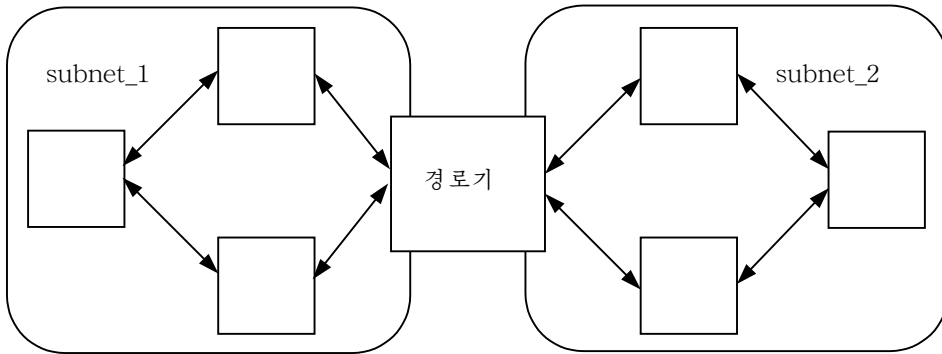
그림 13-13. 두개의 부분망으로 분리
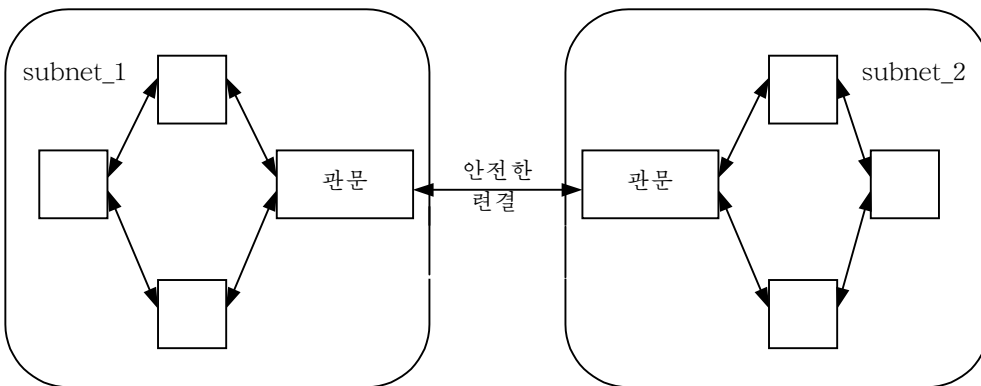
그림 13-14. 가상사설망

- 망에로의 접근을 조종한다.
- 망으로부터 나가는 자료에 암호학적보호를 첨부한다.
- 망의 내부구조를 은폐시킨다.

등에 리용될수 있다. 다음절에서는 망경계에서 실현될수 있는 보안기구들의 형태를 보다 구체적으로 고찰한다.

제4절. 방화벽

우의 실례를 계속하자. 기관의 LAN은 PC들을 련결한다. PC우의 조작체계는 사용자에게 편리하지만 보안을 위하여 설계되여 있지 않다. LAN이 기관의 내부에서 쓰이는 경우에는 보안이 필수적인 문제로는 되지 않는다. 그러나 LAN이 인터네트에 련결될 때에 위협환경이 변한다. 기관은 종업원자료에 대한 보안방책을 확장하기로 결정하고 종

업원자료가 절대로 로동행정부밖에 있는 마디에로 나가지 않도록 요구할수 있다. 로동행정부부분망을 벗어 나는 모든 통신량을 차단하는것은 선택권으로가 아니라 이 보안방책을 시행하는 새로운 기구들을 받아 들이는 방법으로 하여야 한다. LAN의 모든 마디들에서 보호를 개선하는것은 보안의 각도에서 보면 틀린 사고방식이라고 말할수는 없지만 그렇다고 하여 현실적이라고는 볼수 없다.

보안에 최대의 영향을 주면서도 될수록 변화가 적게 되도록 하기 위하여서는 내부망이 외부세계와 접촉하는 위치에 보안기구들을 집결하여야 한다.

정의: 내부망의 경계를 보호하는 임의의 보안체계에 대한 일반적인 이름으로서 《방화벽》을 사용한다. 장벽주체계란 외부세계에 로출된 보안이 강한 콤퓨터체계를 말한다.

방화벽들은 인터네트보안과 밀접히 련결되여 있다. 독자들은 인증규약의 리용방법을 고찰할 때 사용자들이 누구인가를 알고 있어야 한다. 인터네트우에서 계속 늘어 나는 미지의 사용자들을 만나려고 할 때 이 가정은 적합하지 못하다. 만일 미지의 사용자들에게 통제된 접근을 제공하려고 한다면 어떻게 하면 되겠는가? 원격사용자로부터 오는 주소 또는 요구한 봉사는 접근조종을 결정할수 있다. 따라서 방화벽의 기본과제는 다음과 같다.

- 송신자 또는 수신자주소에 기초한 접근조종,
- 요청된 봉사에 기초한 접근조종,
- 내부망(위상배치,주소들, 외부세계로부터 오는 통신량)의 은페,
- 들어 오는 파일에 대한 비루스검사:이것은 전자우편으로 마크로비루스들이 확산되고 있는 비상사태에 특별히 적합하다.
- 통신량의 원천에 기초한 인증,
- 인터네트활동의 체계등록.

방화벽에서 리용하는 기본적인 기구들은 파케트려과와 대리봉사의 두가지 형태가 있다.

1. 파케트려과

망규약은 원천주소로부터 목적주소에로 파케트들을 보낸다.파케트에 관계되는 정보는 자기의 머리부에 있다.이 머리부에는 원천주소와 목적지주소는 물론 파케트가 속하는 《응용규약》을 알려 주는 일부 정보도 포함되여 있다. 실례로 TCP포구번호 23은 Telnet파케트를 식별한다. 파케트려과는 다음의 정보에 기초하여 진행된다.

- **원천주소:** 원천주소들을 쉽게 위조할수 있으므로 용도를 제한한다.즉 내부원천주소를 가지고 오는 인터네트로부터 파케트들이 도착하지 못하도록 해주어야 한다.
- **목적주소:** 장벽주체계로만 파케트들을 보내는 그림 13-16의 차폐하는 경로기가 대표적인 실례라고 볼수 있다.
- **규약:** TCP포구번호를 리용하면 파케트가 속하는 규약에 기초하여 파케트를 려과할수 있다. 실례로 FTP를 허용하고 Telnet를 차단시킬수 있다.

- **련결:** 망층려과(국가적인 조사)는 파케트들을 련결과 결부시키면 실례로 내부 FTP 요구에 응답하여 도착하는 FTP파케트들을 인터네트로부터의 련결에 속하는 파케트들과 구별할수 있다.

2. 대리봉사

여기서도 FTP를 실례로 들어 설명하자. 내부망에 있는 의뢰기가 인터네트우의 봉사기를 호출하려고 할 때 보안방책은 제한된 범위의 사용자들에게만 FTP의 사용을 허락하며 사용자들이 공격적이라고 생각하는 자료들의 내리적재(download)를 금지하게 할수 있다. 파케트는 Telnet련결을 차단하고 FTP를 허용하는 방책들을 지원하고 있지만 여기서는 전혀 도움을 받을수 없다. 더우기 외부 FTP봉사기에로의 접근이 허락되면 의뢰기의 망주소가 공개될것이며 이것은 잠재적인 공격자들에게 유용한 정보로 된다.

사용자신원에 기초하여 방책들을 식별하며 내부망의 정보를 은폐하기 위하여 대리봉사를 사용한다. 대리봉사들은 통제된 호출의 다른 하나의 례라고 볼수 있다. 대리봉사들은 의뢰기요구를 차단하고 자기의 보안규칙들에 맞게 허가되여 있는가를 결정한다. 만일 허가되여 있다면 요구는 실제적인 봉사에로 넘어 간다. 대리봉사기는 외부세계가 알수 있는 유일한 실체로서 내부사용자들에게는 투명한것처럼 보인다. 대리봉사기는 규약에 특정인 접근규칙을 적용하며 사용자신원과 파케트내용에 근거하여(실례로 특수한 명령이 실행되는것처럼 생각할수 있다.) 접근조종을 실현할수 있다. 실례로 대리봉사기는 FTP를 통과시켜 요구를 접수하고 FTP를 차단하여 보류시킬수 있다. 사용자들은 방화벽에 체계를 등록하여서는 안되므로 보관되여야 하는 등록자리는 없다. 응용에 관계되는 사건들은 등록될수 있다. 물론 보호하려고 하는 매개 봉사마다 대리봉사기를 요구할수 있다. 이 방법은 시장에 출하되는 인터네트봉사들이 계속 증가하고 있는 현실에 쉽게 대처할수 없다.

동일한 원리를 어떠한 규약준위에서나 리용할수 있다. 만일 개별적인 응용규약을 보호하려고 한다면 이 규약에서 교환된 통보문(대화)들을 방화벽에서 다시 조립하고 응용에 특정인 보안검사를 적용하여 예정된 봉사와 통신을 진행할것이다. 그러나 복합규약에서는 이러한 전략으로 하여 성능저하가 생길수 있다.

3. 2중홈주콤퓨터방화벽

2중홈주콤퓨터란 대면부가 두개인 기계를 말한다. 이러한 방화벽은 인터네트와 내부망사이에서 파케트들을 단순히 보내는것이 아니라 이 파케트들을 자기의 보안규칙에 따라 처리한다. 2중홈주콤퓨터는 《본질에 있어서 하나의》 방화벽이다. 즉 파케트려파와 대리봉사를 제공한다. 내부망우의 의뢰기들은 방화벽에 있는 대리봉사기들을 리용하거나 방화벽에 직접 등록함으로써 인터네트에 있는 봉사들에 접근할수 있다.

4. 차폐형주콤퓨터방화벽

2중홈주콤퓨터들은 일반적으로 Unix체계에 적응하여 구성된다. 그림 13-15의 방화벽은 외부세계와 직접 마주하고 있는 복합체계로 된다.

이와 같은 환경에 대하여 믿음이 가지 않으면 방화벽이 하는 기능의 일부를 떼내여 기능은 제한되지만 보다 강한 체계를 구축하고 그를 통하여 인터네트에 련결시킬수 있다.
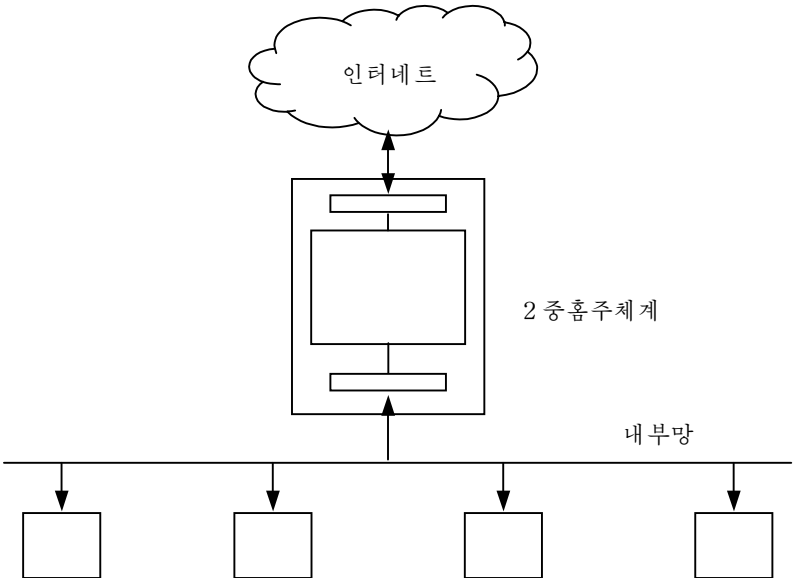
그림 13-15. 2중홈주콤퓨터방화벽

그림 13-16의 방화벽([28]에서는 차폐형주콤퓨터방화벽이라고 부른다.)은 파케트려과를 실현하고 인터네트와 대면부를 제공하는 차페하는 경로기와 내부망우에 있는 장벽주콤퓨터로 이루어 져 있다.
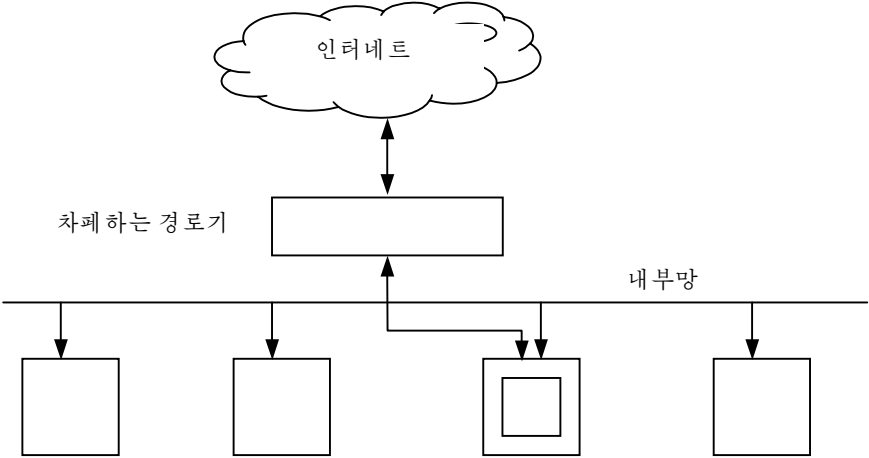
그림 13-16. 차폐형주콤퓨터방화벽

차페하는 경로기는 허용되는 모든 입력통신량을 장벽기계에로 보내며 여기서 파케트들을 내부망으로 보내기전에 다시 접근조종을 결정할수 있으므로 장벽기계에서 오는 내부파케트들만 접수한다.

파케트려과를 방화벽이 실행하는 기타 과제들과 분리시키면 복합경로기의 기능이 단순해 지므로 두가지 성능이 다 좋아 진다. 그것은 경로문제에 대하여 하드웨어를 최량화할수 있으므로 보안에 대한 믿음성이 더 높아 지기때문이다.

이런 의미에서 차페형주콤퓨터방화벽은 2중홈기계에 비하여 안전하다고 주장할수 있다. 그러나 높은 준위 담보는 제한된 기능에만 적용되고 오히려 대리봉사기들이 보안과 관계되는 특징들을 보다 많이 제공한다.

5. 차페형부분망방화벽

차페형부문망은 우에서 본 두가지 방법(그림 3-17)의 특징을 결합하고 있다. 주변망(또는 비무장지대(DMZ)라고도 부른다.)은 내부망과 인터네트사이에 위치하고 있다. 차페하는 경로기는 인터네트와 주변망사이에 놓인다.

외부리용자들을 처리하는데는 단순한 파케트려과방책들이면 충분하다.

주변망과 내부망사이의 2중홈기계방화벽은 믿음성은 떨어 지지만 내부사용자들을 지배하는 보다 복잡한 방책들을 적용할수 있다.
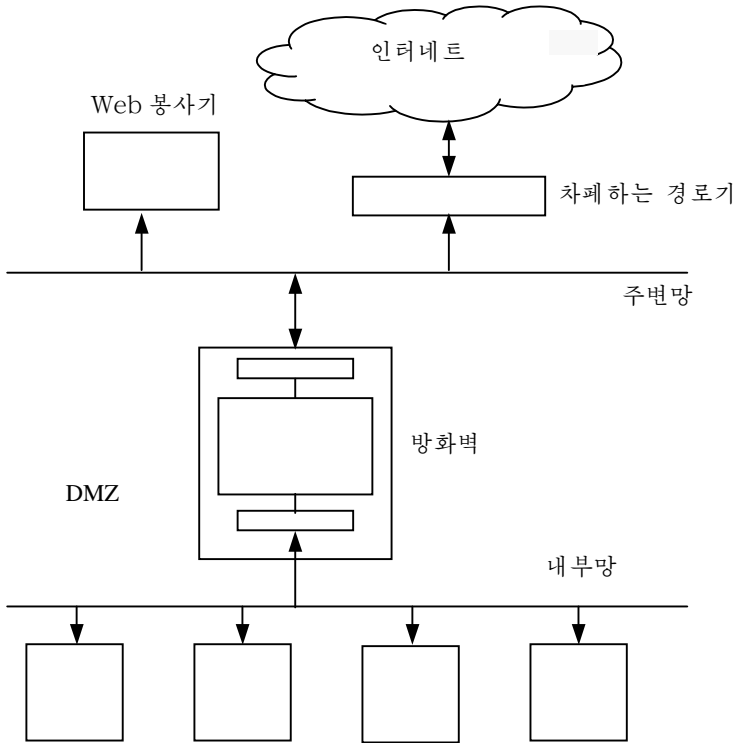
그림 13-7. 차페형부분망방화벽

주변망은 Web봉사기와 같이 외부세계에 호출될수 있어야 하는 민감하지 않은 기계들에 적합한 위치로 된다. 당신의 Web페지를 열람하는 외부사람들은 내부망에 전혀 들어 갈 필요가 없으며 만일 실제적으로 대중적인 신망에 관심을 가진다고 하면 봉사기에 있는 자료를 CD-ROM과 같은 쓰기 불가능한 매체에 기억시켜야 한다.

요약

방화벽의 구축에서 제기되는 문제들은 두가지로 겹쳐 있다. 기능의 측면에서 방화벽의 보안기구들은 주문자보안방책을 만족시켜야 한다. 주문자보안방책은 흔히 주소에 기초한 방책과 신원에 기초한 방책이 혼합되여 있다. 이밖에도 방화벽은 약화되거나 우회될수 없다는 담보가 있어야 한다. 복합체계에서는 기계들이 공격을 받을수 있는 위치에 있게 되므로 이러한 담보에 도달하기가 쉽지 않으며 방화벽을 틀리게 등록하여 침해를 받은 사실들은 많다.

내부망우에 있는 기계의 전화가입선들은 공격자들이 방화벽을 우회하게 하는 고리로 되고 있다. 실현하기 시끄러운 보안시행들은 상반되는 보안시행으로 시험하는 경우도 있다. 만일 방화벽이 지나치게 망접근에 간섭한다면 사용자들은 효과성문제를 들고 나오면서 전화가입선들을 요구할것이다. 끝으로 이야기하고 싶은것은 방화벽이 정확히 설치되여 있다고 하여도 내부기계에 대하여 보안을 하려고 하는 경우에는 차단해 주는것이 더 좋다는것이다. 그 리유는 방화벽은 내부로부터의 공격에 대해서는 보안을 거의 할수 없기때문이다.

이 장의 문헌안내

이 장에서는 망보안에서 제기되는 문제점들과 방법들을 간단히 고찰하였다.
망보안은 [53]에서, 인터네트보안은 [8]에서 포괄적으로 취급되였다.
Internet Engineering Task Force의 Web싸이트는 다음과 같다.

http://www.idtf.org

IETF규격들은 아직 완성되지 못하였다.
이에 대해서는 인터네트보안에 관한 도서들에도 반영되였다.

RFC와 인터네트초안에 대한 자료들도 Internet Draft Stiod on Directories 로부터 얻을수 있다. 즉

ds.internic.ner(US East Coast)
nic.nordn.net(Earope)
ftp.isi.edu(US West Coast)
munnari.oz.an(Pacific Rim)

IPSEC는 RFC 1825, RFC 1826, RFC 1827에서 정의되였다.
SSL우의 Netscape페지는 다음과 같다.

http://www.netscape.com/eng/ss13

방화벽에 관한 가치 있는 도서는 [28,29]이다. [28]에서도 Web보안에서 방화벽의 역할에 대하여 넓은 범위에서 취급하였다.

믿음성 있는 정보체계(TIS)는 쉽게 응용할수 있는 방화벽도구를 가지고 있으며 그의 Web싸이트주소는

http://www.sctc.com

판매자Web싸이트는 현재의 방화벽제품들을 찾아 보기 위한 장소이다.
례하면 사이드와인더방화벽에 관한 자료는 다음의 주소에서 찾을수 있다.

http://www.sctc.com

련습문제

1. ARP(주소결정규약)는 하드웨어주소들을 IP주소들과 련관시키고 있다[8]. 이 련관은 시간에 따라 변한다.망의 매개 마디는 대응하는 IP와 하드웨어주소에 대한 ARP 완충기억기를 가지고 있다.완충기억기의 입력은 몇분이 지나면 유효기간이 끝난다. 자기의 완충기억기에 없는 IP주소에 대한 하드웨어주소를 찾으려고 시도하는 마디는 IP와 하드웨어주소를 포함하는 ARP요구를 알려 준다. 요구한 IP주소를 가지는 마디는 자기의 장치주소로 응답한다.다른 모든 마디는 이 요구를 무시할수 있다. ARP규약에서 기만할수 있는 공간은 무엇인가? 기만에 대해 어떠한 방어대책을 취할수 있는가?

2. IPSEC에서 인증머리부 또는 ESP머리부로 자료본문을 인증할수 있다.두 방법들을 비교하시오.어느것이 더 좋은 보호방법으로 되는가?

3. IPSEC와 SSL에서 규약을 시행하는 마디들은 안전하다고 가정한다.이 가정이 사실로 되게 하기 위해서는 이 마디들에 어떠한 보안기구들을 보충할 필요가 있는가?

4. 안전한 전자우편체계로부터 어떠한 보안특징들을 기대할수 있는가? 안전한 전자우편을 실행하고 있는 기계로부터 기대할수 있는 보안특징들은 무엇인가? 닉명을 제공하려는 봉사와 그렇지 않은 봉사들사이에 차이를 말하시오.

5. 망들은 점차 원격으로 관리되고 있다. 원격망관리에 대한 보안의 의미를 분석하고 안전한 망관리에 요구되는 보안기구들을 서술하시오.

6. 전문가들은 대리봉사기들과 파케트려과방화벽의 상대적강도에 대하여 의견을 달리하고 있다.전문가들의 론거를 하나하나 고찰하고 자기 견해를 세워 론박해 보시오.

7. 인터네트는 비루스감염의 새로운 원천으로 되고 있다.방화벽들은 외부로부터 내부망을 보호한다. 방화벽들이 비루스감염을 막을수 있는가? 여러가지 형태의 방화벽들을 고려하여 대답하시오. TCP/IP층 또는 응용층에서 암호보호가 비루스를 보호하는 방화벽의 능력에 어떤 영향을 주는가?

8. 통신보안에서 MLS방책[129]들의 시행을 고려하여 콤퓨터망에 련결된 요소들로 여러준위보안체계를 구성하시오. 개별적인 마디들의 조작체계에서 어떠한 보안특성들이 요구되는가?

제4편. 리론

제14장. 자료기지보안

자료기지는 단순히 자료를 기억만 하는것이 아니라 사용자들에게 정보를 제공한다. 따라서 자료기지보안은 민감자료의 보호와 관련되여 있을뿐아니라 사용자들이 일정한 통제를 받으면서 정보를 검색할수 있는 기구로 고찰되여야 한다. 이 두가지 문제점들은 조작체계보안과 자료기지보안의 차이를 대조적으로 강조하고 있다. 자료에로의 접근에 비하여 많은 정보에로의 접근을 조종하여야 하므로 접근을 요구하는 주동체의 조종에 초점을 두어야 하는것은 명백하지만 자료의 보호는 여전히 중요한 문제로 되고 있다.

목적

- 자료기지체계에 고유한 보안문제들을 분석한다.
- 관계형자료기지에서 보임새가 어떻게 접근조종에 리용될수 있는가를 리해한다.
- 통계자료기지에서 정보보호문제를 평가한다.
- 자료기지관리체계와 그밑에 놓이는 기본조작체계의 보안기구들사이의 잠재적인 호상작용을 검토한다.

제1절. 소개

자료기지란 어떤 의미를 가지도록 배렬한 자료의 집합을 말한다. 자료기지관리체계(DBMS)는 자료를 편성하여 사용자들에게 정보를 검색할수 있는 수단을 제공한다. 만일 정보에로의 접근을 완전히 조종할수 없다면 어떤 자료를 자료기지에 보관하는것을 아마도 주저하게 될것이므로 자료기지체계의 봉사는 쓸모가 적어 지게 된다. 실례로 자료기지는 일반적으로 개별적인 대상들에 대한 정보(이를테면 회사에서는 종업원들의 기록, 대학에서는 대학생들의 기록, 세관에서는 수입관세목록)들이 보관되여 있다. 여러 나라들에서는 사적비밀보호에 관한 법을 제정하고 있으며 개인자료를 보호하기 위한 법적책임을 지니고 자료기지를 관리하는 기관들을 두고 있다. 따라서 자료기지보안은 초기부터 콤퓨터보안에서 중요한 자리를 차지하였다. 자료기지보안은 조작체계보안과 구별되기때문에 특별한 자리를 차지하였다. 아래에서 이렇게 주장할수 있는 론거를 보여 준다.

조작체계는 자료를 관리한다. 사용자들은 조작체계기능을 호출하여 파일을 창조하거나 삭제 또는 파일을 열어 읽기 또는 쓰기를 한다. 이러한 조작들중 어느 하나도 파일의 내용을 론의하는것은 없다. 파일의 창조나 삭제, 열기는 조작체계에 의한 접근조종결심에 따라 진행되는 가장 적합한 실례로 된다. 접근조종의 결심은 파일의 내용이 아니라

사용자의 신원, 파일에 정의된 허락, 접근조종목록, 보안표식 등에 관계된다. 이것은 어떤 원리적인 보안리론이 아니라 단순히 타당한 공학적인 결정방법에 기초하고 있다.

자료기지의 기입항목은 정보를 담고 있다. 자료기지사용자들은 자료기지 기입항목의 내용을 고려하여 조작을 진행한다. 자료기지의 탐색은 자료기지의 가장 대표적인 리용이라고 말할수 있다. 바로 그렇기때문에 자료기지관리체계에 의한 접근조종을 결정할 때 자료기지의 기입항목의 내용도 고려하게 된다. 보편적인 실례로서 주어 진 한계값보다 높은 생활비들에 대해서는 기밀성이 준수되여야 하는 생활비자료기지를 들수 있다. 요약해서 말한다면 자료기지보안은 사람-기계의 관계에서 사용자쪽에 더 치우치게 된다(그림 14-1).

그림 14-1. 사람-기계척도에서 본 자료기지보호의 위치

얼핏 보기에는 자료기지에서 민감정보를 보호하는것이 쉬운것처럼 생각할수 있다. 생활비자료기지에서는 생활비의 액수를 검사하는 조건을 단순히 질문에 첨가하면 된다. 만일 어느 자료를 보호하여야 하는가를 알고 있다면 이러한 방법은 틀림없이 실현하기 쉽다고 볼수 있지만 침입자가 여러가지 사소한 정보에도 상당히 흥미를 가진다면 사정은 달라 진다. 아래에서는 가능한 정보원천의 범위를 보여 주고 있다.

- **정확한 자료**: 자료기지에 보관된 값.
- **한계**: 생활비와 같이 수값의 상한 또는 하한은 이미 쓸모 있는 정보일수 있다.
- **부정적결과**: 자료기지에 많은 범죄적증거가 포함되여 있다면 개별적사람들이 조금이라도 확신하는 정보는 민감정보이다.
- **존재성**: 자료의 존재는 그자체가 민감한 정보로 될수도 있다.
- **확률값**: 다른 질문의 결과로부터 어떤 정보를 추축할수 있는 실재.

결국 모든 우발적인 사건에 대처하여 자신을 방어하여야 한다. 만일 자료기지가 통계적질문을 허용한다면 정보의 보호는 보다 까다로운 문제로 된다. 실례로 통계적질문으로는 전체 생활비의 합이나 평균값을 돌려 줄수 있다. 이러한 질문들을 잘 결합하면 보호하려고 하는 정보를 알아 낼수 있다. 이러한 문제를 제14장 4절에서 고찰한다.

민감정보가 자기의 자료기지에서 루설될수 있는 여러가지 경로에 대하여 이미 주의를 주었다.물론 보안을 신중하게 선택하여야 하지만 자료기지가 어떤 유용한 목적에 쓰인다는 사실을 놓치지 말아야 한다. 자료에 접근하지 못하게 하는 지나치게 제한적인 방책은 비록 민감정보가 로출되는 일이 없다 하더라도 자료기지체계의 가치는 떨어 진다. 그러므로 민감정보를 보호하면서도 민감정보가 아닌것은 될수록 공개되도록 정밀성을 보장하기 위하여 애써야 한다.

자료기지의 기입항목은 콤퓨터체계의 외부에 있는 실체에 대한 정보를 담고 있다.이러한 실체로서는 창고의 재고수준,대학생들의 시험결과,은행구좌의 잔고량,비행기좌석예

약이 될수 있다. 자료기지기입항목들은 이러한 외부사실들을 정확히 반영하고 있어야 한다. 자료기지보안은 응용에 특정인 완정성보호를 결합하여 다음과 같은 내용들이 결합되여야 한다.

- **내부적일관성**: 자료기지에서 기입항목들은 어떤 제정된 규칙에 따른다. 실례로 재고수준은 령아래로 떨어 질수 없다.
- **외부적일관성**: 자료기지에서기입항목들은 정확하다. 실례로 자료기지에서 나타내는 재고수준들은 창고의 재고수준과 일치한다. 자료기지관리체계는 자료기지를 갱신할때 오유를 피하게 해줄수 있지만 DBMS에만 의거해서는 계속 일관한 상태로 되게 할수 없다. 이러한 속성을 정확성이라고 한다.

제1장 4절의 계층모형에서 자료기지관리체계는 조작체계의 웃부분에 있는 봉사층에 놓여 있을수 있다. DBMS는 조작체계에서 취급하지 않는 자료기지에 고유한 보안요구들을 만족시켜야 한다.DBMS는 조작체계안에서의 보호기구와 결합하여 보안을 시행할수 있으며 또한 조작체계에 적합한 조종기능이 없는 경우 혹은 조종기능이 복잡하여 조작체계에 포함시킬수 없는 경우에는 자기의 조종기능에 기초하여 보안을 실현한다. 더우기 DBMS는 응용층에서 보안조종을 정의하는 도구로도 될수 있다. 그림 14-2는 자료기지보안이 서로 완전히 다른 추상화의 층에 보안기구들을 포함하고 있다는 사실을 보여 주고 있다.

그림 14-2. 자료보호의 위치

제2절. 관계형자료기지

오늘 관계형자료기지는 자료기지를 편성하는 여러가지 모형들중에서 가장 널리 리용되고 있다. 여기서는 자료기지에 대한 개념을 이미 알고 있다고 가정하고 관계형자료기지에 대하여 간단히 소개만 한다. 구체적인 내용은 [37]을 참고하시오.

관계형자료기지: 표(table)와 그것들의 집합으로만 이루어 진 자료기지.

관계형자료기지에 대한 이 정식화는 그것의 물리적인 편성을 의미하는것이 아니라 사용자에 의하여 지각되는것을 의미한다. 관계형자료기지는 또한 자료기지보안을 론의하는데 적합한 추상화준위로 되는 경우도 있다.

관계(relation) R는 형식적으로 $D_1 \times D_2 \times \cdots \times D_n$ 의 부분모임이다. 여기서 $D_1, \cdots, D_n$는 n개의 속성을 가지는 령역이다. 관계에서 요소들은 $v_i \in D_i$인 n개의 무이($v_1, \cdots, v_n$)로 된다. 즉 i 번째 속성의 값은 D_i의 요소로 되게 된다. 무이(tuple)안의 요소들을 흔히 마당(field)이라고 부른다. 마당에 아무런 값도 들어 있지 않다면 이 자리에 특수한 령값(null)을 들여 보내여 이것을 나타낸다. 이 령의 의미는 《기입항목이 없다.》는 뜻을 나타내지 《기입항목이 알려 져 있지 않다.》는것이 아니다.

그림 14-3에서 관계들은 려행사자료기지의 부분으로 될수 있다. 관계 Diary는 이름, 날자, 비행번호, 상태의 4가지 속성을 가진다.

이름	상태
Alice	사사
Bob	공무

이름	날자	비행번호	상태
Alice	월	GR123	사적
Bob	월	YL011	공무
Bob	수	BX201	
Carol	화	BX201	공무
Alice	목	FL9700	공무

비행번호	목적지	출발시간	날자
GR123	THU	7:55	1 - - 4 - - -
YL011	ATL	8:10	1 2 3 4 5 - 7
BX201	SLA	9:20	1 - 3 - 5 - -
FL9700	SLA	14:00	- 2 - 4 - 6
GR127	THU	14:55	- 2 - - 5 - -

그림 14-3. Diary와 Flights사이의 관계

- 이름: 유효한 모든 주문자이름
- 날자: 요일 즉 일, 월, 화, 수, 목, 금, 토
- 비행번호: 2문자로부터 4문자까지로 된 비행번호
- 상태: 공무려행, 사사려행

관계형자료기지에서 정보를 검색하거나 갱신할수 있는 방법을 기술하는 표준언어는 구조화된 질문언어 SQL[52]이다. 자료조작을 위한 SQL조작들에는 다음과 같은 작용이 포함되여 있다.

SELECT: 관계에서 자료를 검색한다.

실례:

```
SELECT Name,Status
  FROM Diary
  WHERE Day='Mon'
```

은 다음과 같은 결과를 돌려 준다.

UPDATE: 관계에 있는 마당을 갱신한다.

실례:

```
UPDATE Diary
  SET Status=private
  WHERE Day=' Sun'
```

은 일요일의 모든 려행이 사사려행이라는것을 나타낸다.

DELET: 관계로부터 무이를 삭제한다.

실례:

```
DELETE FROM Diary
   WHERE Name ='Alice'
```

는 Diary에서 Alice에 대한 려행을 모두 지워 버린다.

INSERT: 무이를 관계에 삽입한다.

실례:

```
INSERT INTO Flights(Flight,Destination,Days)
   VALUES('GR005','GOH','12-45-')
```

는 FIight에 새로운 무이를 삽입한다. 이때 마당 Depart은 아무런 변화도 없다.

모든 경우에 보다 복잡하게 구성할수도 있다. SQL의 복잡한 문제들을 일일이 설명하는것은 이 책의 목적이 아니므로 례증으로 될수 있는 하나의 실례만 제시하겠다. 실례로 Thule로 가려고 하는 사람을 찾기 위하여 다음의 프로그람을 실행시킨다.

```
SELECT Name
   FROM Diary
   WHERE Flight IN
     (SELECT FIight
        FROM Flights
        WHERE Destination='THU')
```

자료관계들은 일반적으로 표로 시각화할수 있다. 속성들은 표에서 렬에 해당하므로 속성의 이름을 그 렬의 표제로 나타낸다. 표의 행들은 자료기지에서 무이(기록)에 대응한다. 관계모형에서 관계에는 련결(relationship)이나 다른 표에로의 지시자가 포함될수 없다. 표(관계)들사이의 련계는 오직 다른 관계에 의해서만 주어 질수 있다.

관계형자료기지에서 여러가지 종류의 관계들이 존재할수 있다.

- **기본관계**: 실관계라고도 부른다. 이름 붙은 관계 및 자률적인 관계이다; 그것들은 그자신의 권한에 속하여 다른 관계로부터 유도되지 않으며 《자기의 고유한》 기억된 자료를 가진다.
- **보임새**: 이름을 가지는 파생관계들로서 다른 이름을 가지는 관계에 의하여 정의된다; 자기의 기억된 자료를 가지지 못한다.

- **순시상**: 보임새와 마찬가지로 이름을 가지는 파생관계들로서 다른 이름을 가지는 관계에 의하여 정의된다; 자기의 기억된 자료를 가진다.
- **질문결과**: 질문의 결과; 그것들은 이름을 가지거나 가지지 못할수 있다. 그것들을 자료기지 per se에서 지속적으로 존재하지 않는다.

실례로 누가 려행하려고 하며 날자가 언제인가를 나타내는 Diary순시상은 다음과 같이 정의한다.

```
CREATE SNAPSHOT Travellers
    AS SELECT name.day
    FROM Diary
```

1. 자료기지열쇠

매 관계에서 전체 무이들은 유일한 방법으로 식별할수 있어야 한다. 때때로 하나의 속성이 식별자로 리용될수 있다. 이 목적에 쓸수 있는 속성들의 선택이 있을수 있다. 한편 하나의 식별자를 구성하는데 하나이상의 속성이 필요되는 경우도 있을수 있다.

정의: 관계의 1차열쇠는 그 관계에서 하나밖에 없는 최소식별자이다. 관계 R의 1차열쇠 K는 다음의 조건을 만족시켜야 한다.
- **일값성**: 임의의 시각에 K에 대하여 같은 값을 가지는 R의 무이는 없다.
- **최소성**: K가 합성열쇠이면 일값성을 파괴하지 않으면서 생략할수 있는 K의 성분은 없다.

우의 례 관계 Diary에서 이름과 날자의 조합은 1차열쇠로 될수 있다(주문자들이 하루에 한번의 려행만 한다고 가정한다). 관계 Flights에서 1차열쇠는 비행번호이다.

모든 관계는 1차열쇠를 반드시 가져야 한다. 그것은 2중화되는 무이를 포함할수 있는 관계는 없기때문이다. 이것은 관계의 형식적인 정의로부터 직접 나온다. 어떤 관계의 1차열쇠가 다른 관계의 속성으로 쓰일 때 이것을 그 관계의 외부열쇠라고 부른다. 우의 실례에서 비행번호는 관계 Flights에서 1차열쇠로 되지만 관계 Diary에서는 외부열쇠로 된다.

2. 완정성규칙

관계형자료기지에서는 완정성규칙들을 정의하면 내부일관성을 실현하고 외부일관성(정확성)을 유지하는데 도움이 될수 있다. 대부분의 완정성규칙들은 응용에 따라 다르지만 관계형자료기지모형에 고유한 규칙들은 두가지이다.

실체완정성규칙: 기본관계에서 령을 접수할수 있는 1차열쇠의 요소는 없다.

이 규칙은 우리들이 기본관계들에서 전체 무이를 찾을수 있게 한다. 기본관계에서 무이들은 《실제적인》 실체에 대응하기때문에 우리가 그것을 식별할수 없다면 자료기지에서 그러한 식별을 표현하지 못할것이다.

참조완정성규칙: 자료기지는 맞지 않는 외부열쇠값들을 포함하고 있지 말아야 한다.

외부열쇠값은 어떤 다른 표의 기입항목에 대한 참조를 나타낸다. 맞지 않는 외부열쇠값이란 참조된 표에서 1차열쇠로 나타나지 않는 값을 말한다. 맞지 않는 외부열쇠값은 존재하지 않는 무이의 참조로 된다.

이 두가지 규칙외에도 응용에 특정인 완정성규칙들이 더 있을수도 있으며 이 규칙들에 의해 자료기지가 쓸모 있는 상태로 되기때문에 중요하다고 말할수 있다. 대표적으로 이러한 완정성규칙들을 리용하여 다음과 같은것을 할수 있다.

- **마당검사**: 자료기입항목에서 오유가 발생하지 못하게 한다. 실례로 Diary관계에서는 기입된 값이 공무려행인가 사사려행인가를 검사하는 규칙을 통하여 상태(status)속성에 임의의 값들이 삽입되지 않도록 한다.
- **유효범위검사**: 통계자료기지에서는 질문의 결과가 충분히 많은 표본에서 계산되였는가를 검사하는 규칙이 있으면 좋을것이다. 그림 14-4의 Student관계를 본다면 표본의 크기가 3보다 크지 않는 경우에는 평균성적 67을 기정으로 되돌리는 규칙을 정의할수 있다.
- **일관성검사**: 서로 다른 관계에서 기입항목들은 외부세계의 동일한 측면을 가리킬수도 있으므로 이 측면에 대한 일관성 있는 견해를 표현하여야 한다. 실례로 주문자가 려행하는 날자가 예정된 출발날자인가를 검사할수 있다. Alice가 GR123항로를 월요일에 려행한다는것은 이 항로가 월요일과 목요일에만 출발한다는 사실과 모순되지 않는다. 그러나 Carol이 BX201 항로로 화요일에 예약하면 비행기가 월요일, 수요일, 토요일에 떠난다는 사실과 맞지 않는다. 매개의 마당을 비교하는 완정성규칙을 적용하면 려행자가 이러한 오유를 범하지 않도록 할수 있다.

이러한 형태의 완정성규칙들은 응용층에서 관리된다. DBMS는 이러한 규칙을 제정하고 실현하는 하부구조를 제공한다. 실례로 완정성방아쇠는 자료기지에서 대상에 첨부될수 있는 프로그람으로서 그 대상의 개별적인 완정성특성들을 검사한다. UPDATE, INSERT, DELETE조작이 대상을 변경하려고 시도할 때 완정성방아쇠는 절환되여 완정성검사를 한다.

앞으로 기밀성과 완정성사이의 잠재적인 대립문제를 언급하지 않으면서 이 문제에 대한 론의를 계속해 나가려고 한다. 완정성규칙의 평가에서 민감정보의 접근을 필요로 할 때 민감정보를 보호하기 위하여 규칙을 불완전하게(그리고 정확치 못하게) 평가하거나 자료기지의 일관성을 보장하기 위하여 일부 민감정보를 루실하든가 하는 이률배반관계에 부딪치게 된다.

제3절. 접근조종

민감정보를 보호하기 위하여 DBMS는 사용자들이 자료기지를 리용하는 방법을 조종하여야 한다. 조종을 실현하는 방법을 보기 위하여 자료기지의 접근이 두 준위에서 진행된다고 하자.

- 기본관계우에서 자료처리조작,
- 보임새 또는 순시상과 같은 합성조작.

제1장 4절 1로 되돌아 가면 두가지 방향에서 접근조종을 고찰할수 있다.

- 적용할수 있는 조작을 사용자에 제한한다. 또는
- 매개 개별적자료항목에 대한 보호요구를 정의한다.

DBMS에서 합성조작들에 대한 조종은 사용자들이 자료기지를 사용하는 방법을 규제한다. 한편 기본관계의 조작을 검사하면 자료기지의 기입항목들을 보호할수 있다. 조종하려고 하는 접근조작의 형태를 결정함으로써 시행되여야 할 방책의 초점에 영향이 미치게 할수도 있다. 반대로 방책의 초점에서는 조종하는 조작이 어떤 형태인가를 알려 주어야 한다. 어떤 선택항목을 선택해도 다음의 두가지 속성이 있게 된다.

- **완전성**: 자료기지의 모든 마당들이 보호된다.
- **일관성**: 자료항목의 접근을 지배하는 규칙들이 충돌하는 경우는 없다.

만일 각이한 접근조종을 결정하는 여러가지 방법으로 접근될수 있는 자료기지의 요소가 없다면 보안방책은 일관적이다.

합법적인 접근요구를 막아서는 안되며 또한 제정된 접근방책을 우회하는 방법이 있어도 안된다.

1. SQL보안모형

기본SQL보안모형은 이미 익숙되여 있는 형식에 기초하고 있으며 3가지 실체에 기초하여 자유접근조종을 실현한다.

- **사용자**: 자료기지의 사용자들. 가입시 사용자신원을 인증한다. DBMS는 자기의 가입을 실행하거나 조작체계에 의하여 인증된 사용자신원을 접수할수 있다.
- **동작**: SELECT, UPDATE, DELETE, INSERT
- **대상**: 표들, 보임새, 표와 보임새의 렬(속성). SQL에는 또한 사용자가 정의하는 구축자도 들어 있다.

사용자가 객체에 동작을 호출하면 DBMS는 요청된 동작을 허락하겠는가를 결정한다. 객체가 창조되면 사용자는 그의 소유자로 지정되며 초기에 소유자만이 객체에 접근한다. 다른 사용자들은 먼저 특권을 받아야 한다. 특권요소들은 다음과 같다.

(grantor, grantee, object, action, grantable)

SQL보안모형을 뒤받침해 주고 있는것은 특권과 보임새이며 응용지향의 보안방책을 정의하기 위한 틀거리를 제공한다.

2. 특권의 수여와 취소

SQL에서 특권들은 GRANT 와 REVOKE 조작에 의하여 관리된다. 특권이란 특수한 동작이라는 뜻으로서 표의 일정한 속성으로 제한될수 있다. 실례로 두개의 려행사 Art와 Zoe가 Diary표의 부분을 조사 및 갱신하려고 한다.

```
GRANT SELECT, UPDATE(Day,Flight)
          ON TABLE Diary
         To Art,Zoe
```

특권은 선택적으로 취소될수 있다.

```
REVOKE UPDATE
        ON TABLE Diary
        FROM Art
```

다음특징은 이밖에도 GRANT선택항목에 의하여 SQL에서 실현되는 특권을 허락하는 권리를 부여하는것이다.
례를 들어

```
GRANT SELECT
        ON TABLE Diary
        To Art
        WITH GRANT OPTION
```

으로 하면 려행사 Art는 또한 Diary표우의 특권을 Zoe에 줄수 있다.

```
GRANT SELECT
        ON TABLE Diary
        To Zoe
        WITH GRANT OPTION
```

표 Diary의 소유자가 Art에 부여된 특권을 취소하는 경우 Art가 허락하는 모든 특권들은 취소하여야 하며 따라서 취소는 계단식으로 되여야 하며 포기할 필요가 있는 정보는 자료기지체계에 의하여 유지되여야 한다.
또한 언급되여야 할것은 다른 사용자들이 자료의 접근을 일단 허락하였다면 원본자료를 어느 정도 조종할수 있다고 해도 자료의 소유자는 이 자료로부터 도출되는 정보가 어떻게 쓰이게 되겠는가를 조종할수 없다. 원본표에로의 어떠한 《쓰기》접근을 요구하지 않고도 표로부터 자료를 읽고 이 자료들을 다른 표에로 복사할수 있다.

3. 보임새에 의한 접근조종

보임새들은 파생관계들이다. 보임새를 창조하는 SQL조작은 다음의 형식을 가진다.

```
CREATE VIEW view_name[(column[,column]...)]
        AS subquery
        [WITH CHECK OPTION];
```

기본관계의 기입항목에 대하여 직접 특권을 부여함으로써 관계형자료기지에서 접근조종을 실현할수 있다. 그러나 적지 않은 보안방책들은 보임새로 그리고 바로 그 보임새들의 특권으로 나타낼수 있다. 보임새의 정의에서 부분질문은 매우 복잡한 접근조건을 서술할수 있다.
간단한 실례로 Diary 관계에서 모든 공무려행을 포함하는 보임새를 구성한다.

```
CREATE VIEW business_trips AS
        SELECT *FROM Diary
        WHERE Status='business'
        WITH CHECK OPTION;
```

보임새에 의한 접근조종은 응용층의 적당한 장소에 놓일수 있다. DBMS는 조종을 실현하는 도구들만 제공한다. 보임새들이 주의를 끌고 있는 여러가지 리유는

- 보임새들은 유연성이 있고 응용요구에 가까운 서술준위에서 접근조종을 정의할수 있게 한다.
- 보임새들은 문맥의존 및 자료의존보안방책을 시행할수 있다.
- 보임새들은 통제된 호출을 실현할수 있다.
- 안전한 보임새들은 보안표식들을 교체할수 있다.
- 자료를 쉽게 다시 분류할수 있다.

의 특성이 있기때문이다.

그림 14-4의 응용지향접근조종은 다음과 같은 보임새로써 표현할수 있다.

```
CREAT VIEW High_FLyers AS
      SELECT *FROM Students WHERE Grade>
      (SELECT Grade FROM Students WHERE Name=current_user( ));
```

는 보임새를 리용하여 평균성적이 개인의 성적에 비하여 높은 대학생들만 현시한다.

```
CREATE VIEW My_Journeys AS
      SELECT *FROM Diary
      WHERE Customer=current_user();
```

는 보임새를 사용하여 주문자가 예약한 그림 14-3의 려행만 현시한다.

자료기지에 접근조종표를 첨부하면 자유접근조종을 실현할수 있다. 보임새는 이 관계를 가리킬수 있다. 이런 식으로 접근권한을 주고 취소하는 사용자의 권리를 통제하는 방책들은 물론 그룹성원자격에 기초한 접근조종도 표시할수 있다. 이밖에도 보임새들은 보안표식들을 정의하거나 가리킬수도 있다. 실례로 Thule에로 공무려행을 보임새에 의해 다음과 같이 창조함으로써 기밀성을 뚜렷하게 해줄수 있다.

```
CREAT VIEW Flights_ >_CONFIDENTIAL AS
        SELECT *FROM Diary
        WHERE Destination='THU' AND Status='business';
```

보임새를 통한 읽기접근을 조종하는것은 보안방책을 정확히 포착하는것과는 다른 특수한 기술적문제는 제기하지 않는다. 보임새들이 INSERT 또는 UPDATE조작으로 자료기지에 정보를 쓰기하는 정황은 여러가지이다. 첫째로 보임새들에는 대응하는 기본관계의 완전성을 관리하는데 필요한 정보가 없으므로 갱신할수 없는 보임새들이 있다. 실례로 고찰하는 기본관계의 1차열쇠를 포함하지 않는 보임새는 갱신에 리용될수 없다. 둘째로 보임새가 갱신할수 있다고 하여도 관심사로 되는 보안문제들이 몇가지 있다. business_trip를 통해서만 Diary자료기지를 호출하는 려행봉사국은 그림 14-5의 표에 기입되여 있는 지료를 본다. 려행봉사국은 다음의 조작으로 보임새를 갱신할수 있는가.

```
UPDATE business_trips
 SET Status ='private'
 WHERE Name='Alice' AND Day='Thu'
```

이름	성별	요강	단위	평균점수
Alma	F	MBA	8	63
Bill	M	CS	15	58
Carol	F	CS	16	70
Don	M	MIS	22	75
Errol	M	CS	8	66
Flora	F	MIS	16	81
Gala	F	MBA	23	68
Homer	M	CS	7	50
Igor	M	MIS	21	70

그림 14-4. 대학생관계

이름	날자	비행번호	상태
Bob	월	YL011	공무
Carol	화	BX201	공무
Alice	목	FL9700	공무

그림 14-5. 보임새의 실례

이때 Alice에 해당하는 입구는 보임새에서 떨어 져 나간다. 사실상 이 경우에 보임새의 정의에서 CHECK선택항목을 지정하였기때문에 갱신은 허용되지 않을것이다. 만일 보임새의 정의에 WITH CHECK OPTION이 들어 있다면 UPDATE와 INSERT는 보임새의 정의를 만족시키는 자료기지의 기입항목들만 쓰기할수 있다. CHECK선택항목이 생략되였다면 맹목적인 쓰기가 가능하다.

보임새는 SQL보안모형에서 객체로 될뿐아니라 강령으로 볼수도 있다. 보임새를 호출하는 사용자의 특권보다도 보임새의 소유자의 특권에 의하여 보임새가 평가된다면 통제된 호출을 실현하는 또 하나의 방법을 가진다.

보임새에서 접근조건을 SQL의 한계안에서 지정하여야 하며 이것이 너무 제한적이다고 보아 지는 경우에는 보다 표현적인 언어로 작성된 쏘프트웨어패키지(기억되여 있는 수속)들이 자료기지의 통제된 접근을 제공하는 DBMS의 선택항목으로 된다.

소유자의 특권으로 동작하는 이 파케지우에서 특권의 실행이 사용자들에게 또다시 허용된다.

통제된 호출은 콤퓨터체계의 임의의 준위에서 찾아 볼수 있다. 통제된 호출은 자료기지관리체계에서와 마찬가지로 극소형처리기에서도 쓸모 있는 원리로 되고 있다.

지금까지는 보임새가 쓸모 있는 보안기구로 된다는 측면들만 제시하였다. 자연히 보임새들은 약점들도 있다.

- 접근검사로 하여 복잡해 지게 되며 속도가 떨어 질수 있다.
- 보임새의 정의에서 《정확성》이 검사되여야 한다. 보임새들은 목적하는 보안방책을 얻을수 있는가?
- 완전성과 일관성은 스스로 실현되지 않으므로 보임새들을 중첩시킬수 있으며 또는 전체 자료기지를 획득하지 못할수도 있다.
- DBMS(TCB)에서 보안에 관계되는 부분은 상당히 많아 진다.

보임새들은 《표준적인 상업》환경에서는 적합하다. 보임새들은 응용에 맞게 작성될수 있으므로 DBMS를 변경시킬 필요는 없다. 그러면 보임새들의 정의는 업무요구를 가장 적합하게 만족시키도록 자료기지의 구조를 정의하는 일반적인 처리의 한부분으로 볼수 있다.

그러나 접근하려는 개별적자료항목을 결정하는것은 어려운 문제로 된다. 그러므로 사용자의 작용을 조종하기보다는 자료항목을 보호할 필요가 있다고 보아 지는 환경에서는 보임새가 적합치 못하다고 볼수 있다. 제15장에서는 자료기지보안을 자료항목의 보호에 중점을 두고 론의한다.

제4절. 통계적자료기지보안

통계자료기지들은 지금까지 이 책에서 거의나 취급하지 않은 보안문제들을 제기한다. 통계자료기지는 정보가 표의 속성(렬)에 대하여 통계적(접합체)질문에 의하여 검색된다는데 다른 자료기지와 그 특징이 구별된다.

SQL에서 집합체함수들은 다음과 같다.

COUNT:	렬에 있는 값들의 번호,
SUM:	렬에 있는 값들의 합,
AVG:	렬에 있는 값들의 평균,
MAX:	렬에서 제일 큰 값,
MIN:	렬에서 제일 작은 값.

통계적질문에서 질문술어(query predicate)는 대표값을 계산하는데 쓰이게 될 무이들을 지정하며 질문모임은 질문술어와 일치하는 무이들을 의미한다.

한마디로 말하여 통계자료기지에서는 다음과 같은 보안문제가 제기된다.

- 자료기지에는 개별적으로 민감한 자료가 들어 있다. 따라서 자료항목의 직접적인 접근은 허용되지 않는다.
- 자료기지에 대한 통계적질문이 허용된다. 이 질문들은 개별적인 자료항목을 읽는다.

문제를 이와 같이 설정하면 정보를 추론할수 있게 되며 접근요구를 개별적으로 다루는것이 더는 충분하지 못하다는것을 보여 주게 될것이다. 또한 정보흐름에 대하여 보다 실용적인 견해도 있다. 제4장에서 본 기밀모형은 모든것을 다하여 어떠한 정보흐름도 모두 멈춰 세우려고 하였다. 통계자료기지에서는 자료로부터 집합체에로 어떤 정보흐름이 반드시 있게 되며 우리는 그것을 접수할수 있는 준위에로 줄이려고 할뿐이다.

그림 14-4의 대학생자료기지는 이 절의 실례들을 준다. Units와 Grade Ave렬의 개별적기입항목들은 제외하고 모든 속성에 대한 통계적질문이 허용되지만 직접 읽을수 없다. 다음의 통계적질문

```
Q1:SELECT AVG(Grade Ave.)
     FROM Students
     WHERE Programme='MBA'
```

는 전체 MBA대학생들에 대한 평균성적을 계산한다. 이 실례에서 질문술어는 다음과 같다.

```
Programme='MAB'
```

1. 집합과 추론

통계적자료기지보안에서는 집합(Aggregation)과 추론(Reference)이라는 두가지 중요한 개념이 있다. 집합이란 자료기지에 있는 한묶음의 값들에 대하여 계산된 집합체의 민감준위가 개별적인 요소들의 민감준위와 차이날수 있다는 관측을 가리킨다. 집합체의 민감준위가 개별적인 요소들의 준위보다 낮은 경우는 대단히 많다. 이 반대인 경우는 집합체가 덜 민감한 업무자료로부터 얻어 진 민감실행정보인 때일것이다.

집합체는 자료기지에서 또 다른 관계 례하면 보임새로 되므로 이 장에서 제안된 보안기구를 리용하여 집합체에로의 접근을 조종할수 있다. 그러나 공격자는 민감준위에서 차이점을 리용하여 보다 민감한 항목에로의 접근을 얻을수 있다. 추론문제란 비민감자료로부터 민감정보를 이끌어 내는것을 말한다. 다음형태의 공격들을 고찰하여야 한다.

- **직접공격**: 집합체값을 적은 표본에 대하여 계산하여 개별적자료항목의 정보가 루실되도록 한다.
- **간접공격**: 이 공격에서는 여러가지 집합체들에 관계되는 정보를 결합한다.
- **추적자공격**: 특별히 효과적인 형태의 간접공격.

- **선형체계약점**: 추적자는 질문모임들사이에 대수적관계를 리용하여 요구하는 정보가 얻어 지는 방정식을 구성할수 있도록 추적자공격을 선택한다.

2. 추적자공격

이제 그림 14-4에서 제시한 실례의 대학생관계로부터 민감정보를 이끌어 내기 위하여 통계적질문을 리용하는 방법을 설명한다. Garol이 녀성 CS대학생이라는 사실을 알고 있다고 하자. 다음과 같은 합법적인 질문을 결합한다.

```
Q1: SELECT COUNT(*)
     FROM Students
     WHERE Sex='F' AND Programme='CS'
Q2: SELECT AVG(Grade Ave)
     FROM Students
     WHERE Sex='F' AND Programme='CS'
```

Q1로부터 자료기지에는 오직 한명의 녀성CS대학생이 있으며 따라서 Q2가 되돌리는 값 70은 정확히 그의 평균성적이라는것을 알수 있다. 여기서 문제로 되는것은 바로 하나의 모임에 오직 하나의 요소만 포함되여 있다고 선택기준을 규정한것이다. 따라서 통계적질문이 충분히 큰 부분모임을 포함할 때에만 허용해 줄수 있다.

그러나 선택기준을 부정하여 보모임에 간단히 질문하면 전체 자료기지에 적용한 질문의 결과와 우리가 실제로 관심을 가지는 모임의 보모임에 적용한 질문의 결과의 차이로부터 앞에서와 같은 결과를 얻을수 있다. 따라서 질문에서 고려하는 무이의 모임들뿐아니라 그의 보모임도 충분히 크게 해주어야 한다.

유감스럽게도 이렇게 해주어도 충분히 좋은 결과가 얻어 지지는 않는다. 매개 질문모임과 그의 보모임이 적어도 3개의 요소를 포함하여야 한다고 가정하자. 질문순서렬

```
Q3: SELECT COUNT(*)
     FROM Students
     WHERE Programme='CS'
Q4: SELECT COUNT(*)
     FROM Students
     WHERE Programme='CS' AND Sex='M'
Q5: SELECT AVG(Grade Ave)
     FROM Students
     WHERE Programme='CS'
Q6: SELECT AVG(Grade Ave)
     FROM Students
     WHERE Programme='CS' AND Sex='M'
```

은 Q3:4, Q4:3, Q5:61, Q6:58 을 되돌린다. 모든 질문들에서 충분히 큰 무이의 모임을 고려하였음으로 금지되지 않는다. 4개의 결과를 조합하면 Carol의 평균성적은 4·61-3·58=70으로 계산된다.

이 경우 질문의 모임을 이와 같이 구성할수 있었다.

이제부터 체계적인 방법으로 어떻게 공격을 설정하는가를 고찰하자.우선 추적자가

필요된다.

정의: 단일한 무이에 대한 정보를 추적해 낼수 있게 하는 질문술어 T를 그 무이에 대한 개별추적자라고 부른다. 일반추적자란 받아 들일수 없는 임의의 질문에 대한 대답을 찾는데 리용될수 있는 술어를 말한다.

T가 일반추적자이고 R는 조사하려고 하는 무이 r를 일의적으로 식별하는 술어라고 하자. 이 실례에서 술어는 Name='Carol'으로 된다. 술어 R ∨ T와 R ∨ NOT(T)를 리용하여 자료기지에 대한 두가지 질문을 만들어 내자. 우리의 목표 r는 두가지 질문에서 리용되는 유일한 무이로 된다. 두가지 질문이 확고히 접수되게 하기 위하여 질문모임과 그의 보모임을 충분히 크게 선택하여 질문이 허용되도록 T를 선택한다. 전체 자료기지에 대하여 마지막까지 질문하면 공격을 완성하기 위한 모든 자료를 얻게 된다. 이 레에서

Sex='F' AND Programme='CS'

는 Carol에 대4한 개별적추적자로 되며 Programme='MIS'는 일반추적자들중의 하나의 공격자를 나타낸다.

다음과 같이 질문을 계속한다.

```
Q7:  SELECT SUM(Units)
     FROM Students
     WHERE Name='Carol' OR Programme='MIS'
Q8:  SELECT SUM(Units)
     FROM Students
     WHERE Name='Carol' OR NOR (Programme='MIS')
Q9:  SELECT SUM(Units)
     FROM Students
```

이때 Q7:75, Q8:77, Q9:136을 받는다.

따라서 Carol 에는 (75+77)-136=16 단위가 넘어 가게 된다. 경험은 거의 모든 통계자료기지들이 일반추적자를 가진다는것을 보여 준다.

3. 대응책

통계적추론공격에 대한 분석방법은 자료기지보호에 대한 초기 문헌에서 기본으로 되여 있었다. 그이후에 연구자들은 다른 분야에로 주의를 돌리기 시작하였다. 그것은 연구자들이 완성되고 완벽한 해결방도를 찾고 실현하였다고 해서가 아니라 추론공격에 대처하는 대응책에 대한 한계를 인정하였기때문이다. 그 한계가 알려 져 있다면 추론문제에 대하여 현실적으로 무엇을 할수 있겠는가?

먼저 명백하다고 보아 지는 민감정보를 억제해 줄수 있을것이다. 이것은 어느 정보가 민감정보로 되며 어느 민감정보를 억제해 주겠는가를 알고 있으며 따라서 이 정보를 얻어 내는 방법도 알고 있다는것을 암시하고 있다. 적어도 통계적질문의 결과를 공개하기전에 질문모임의 크기를 검사한다.

다음으로 자료를 위장할수 있다. 설사 통계적질문이 정확한 결과를 준다고 하여도 개별적질문이 틀린 결과를 주도록 자료기지의 기입항목을 우연적으로 교체할수 있을것이다. 또한 작은 우연섭동을 질문결과에 첨부해 주어 되돌린 값이 실제값에 가깝지만 그리 정확하지 않게 해줄수도 있을것이다. 이 방법들의 결합은 정밀성과 리용성에 있다.

자료기지방안의 설계에 주의를 돌리면 일부 집합문제들은 쉽게 해결될수 있다[90]. 자료기지구조에 대하여 정적분석은 속성들사이의 민감한 관계를 알아 낼수 있다. 이러한 속성들은 별도로 존재하는 표에 놓이게 된다. 한개의 표에만 접근하는 사용자는 결코 속성들을 상관시킬수 없다. 물론 관련되는 모든 표에 접근하는 사용자에게도 이렇게 할수 있지만 특권을 배정할 때 자료기지관리자는 보다 정확하게 된다. 우리의 실례에서 이름과 기능사이의 관계는 민감하다고 볼수 있다. 대학생표를 두개의 표로 가르고 대학생식별번호와 련결시킨다(그림 14-6).

이제 첫번째 표를 충분히 높은 준위에서 분류하여 우선권이 부여된 사용자들만이 이름과 대학의 기능을 련결시킬수 있다.

끝으로 한가지 질문을 많이 하는것보다도 여러가지 질문을 교묘하게 결합할 때 생기는 추론문제를 관찰해 보면 사용자가 무엇을 알고 있는가는 추적할수 있다. 이것이 가장 좋은 보안으로 될수도 있지만 가장 비용이 많이 드는 방법이기도 하다. 사용자행동을 검열일지에 기록하였다가 의심스러운 질문렬이 검출되였는가를 보기 위해 질문분석수행에 들어 간다. 물론 처음에 무엇이 의심스러운 행동으로 되는가를 알고 있어야 한다. 보호를 더욱 공고히 하기 위하여 질문분석에서는 두명의 사용자 또는 사용자들의 집단이 무엇을 알고 있는가를 고려하여야 할것이다.

이름	ID
Alma	B13
Bill	C25
Carol	C23
Don	M38
Errol	C12
Flora	M22
Gala	B36
Homer	C10
Igor	M20

ID	성별	요강	단위	평균성적
B13	F	MBA	8	63
C25	M	CS	15	58
C23	F	CS	16	70
M38	M	MIS	22	75
C12	M	CS	8	66
M22	F	MIS	16	81
B36	F	MBA	23	68
C10	M	CS	7	50
M20	M	MIS	21	70

그림 14-6. 대학생자료를 위한 몇개의 표들

제5절. 조작체계와의 통합

조작체계의 위치에서 자료기지를 보면 일련의 조작체계처리들과 자료기입항목를 보관하는 기억자원들을 보게 될것이다. DBMS는 여러 측면에서 조작체계와 사명이 류사하다고 볼수 있다. 즉 사용자들이 서로 간섭하지 못하게 하여야 하며 DBMS와도 간섭하지 못하게 하여야 한다.

이 과제들을 조작체계처리에 주면 다시 품을 들이지 않아도 될것이다.

이렇게 설정하면 DBMS는 조작체계처리들의 모임으로서 동작한다.

일반적인 자료기지관리를 위한 체계처리들이 있으며 매개 자료기지사용자는 개별적인 조작체계처리에 대응된다(그림 14-7). 이제 조작체계는 사용자들을 구별할수 있으므로 매개 자료기지객체를 그자신의 파일에 기억시킨다면 조작체계는 모든 접근조종을 할수 있다. DBMS는 사용자질문을 조작체계가 리해하는 조작으로 변환하기만 하면 된다.

개별적인 조작체계처리를 매개 자료기지사용자에게 배정하면 기억자원이 랑비되고 사용자수를 대폭 증가시킬수 없으므로 여러 사용자들의 자료기지요구를 조종하는 처리들이 필요하다(그림 14-8).

이때 기억기는 절약되지만 접근조종의 책임은 DBMS에 넘어 간다.

마찬가지로 자료기지객체들의 기억기에도 이와 같은 고찰방법을 적용할수 있다. 객체들이 너무 적다면 매 객체가 개별적으로 파일을 가지게 하는것은 랑비로 된다. 조작체계가 자료기지사용자들에 대하여 접근조종기능이 없는 조건에서는 하나의 조작체계파일에 마음대로 여러개의 자료기지객체를 묶어 놓을수 있다.
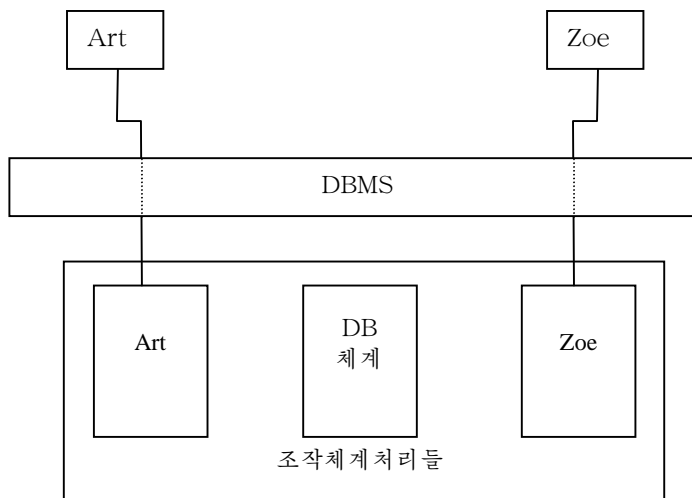
그림 14-7. 조작체계에 의한 자료기지사용자들의 분리
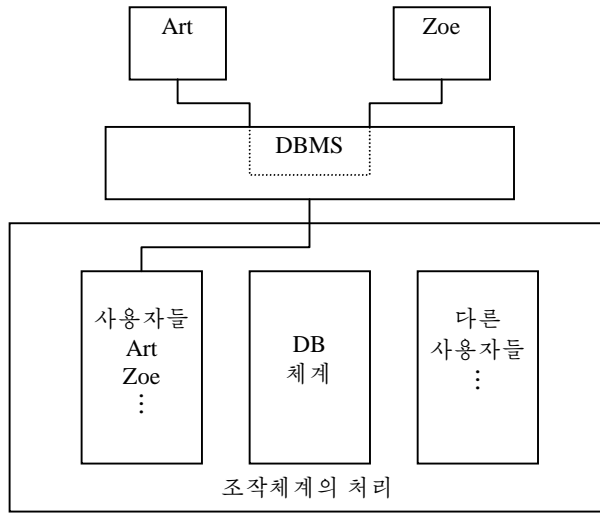
그림 14-8. DBMS에 의한 자료기지사용자들의 분리

이 장의 문헌안내

관계형자료기지모형을 연구할 필요가 있는 경우에는 문헌[37]를 참고하시오. 자료기지보안에 대한 실제적인 자료들은 [25]에서 편집되였다. 초기부터 지금까지 쓸모 있다고 볼수 있는 도서는 [39]이며 이것은 특히 통계자료기지의 좋은 참고서로 된다. 자료기지보안에 대한 쓸모 있는 원서는 [90]이다.

모든 주요자료기지판매자들은 자기의 제품에 대한 정보와 자료기지보안의 좋은 기초를 가진 Web페지를 유지하고 있다. C2준위에서 평가된 제품들을 가지는 자료기지판매자들의 Web페지는 다음과 같다.

http://www.informix.com
http://www.oracle.com
http://www.sybase.com

련습문제

1. 기록(주문자이름, 구좌번호, 잔고, 예금)과 사용자(주문자, 직원, 관리인)로 되여 있는 은행자료기지체계를 생각하자. 접근구조를 실례로 즉 보임새를 통하여 다음과 같이 되게 정의하시오.

   - 주문자들은 자기의 구좌를 읽을수 있다.
   - 직원들은 예금을 제외한 모든 마당을 읽고 모든 구좌에 대한 잔고를 갱신할수 있다.
   - 관리인들은 새로운 기록을 창조하고 모든 마당을 읽고 모든 구좌에 대하여 예금을 갱신할수 있다.

2. 학생의 이름들과 교육강령에서 제시된 모든 과정안들의 표식이 들어 있는 대학생기록으로 된 자료기지를 고찰하시오. 강의자는 강의안들이 아직 종이에 표식되지 않은 과정안에서 모든 대학생들을 보여 주는 보임새를 공급 받는다. 이 보임새는 WITH CHECK OPTION을 정의하여야 하는가? CHECK OPTION을 사용하겠는가를 결정하는 일반평가기준을 말해 보시오.

3. 대학생관계(그림 14-4)에 대한 모든 통계적질문에는 질문모임에 적어도 3개의 무이가 있어야 한다. 속성 Grade Ave에 대한 AVG질문들만 허용된다. 새로운 일반추적자를 찾고 Homeer의 평균성적에 대한 추적자공격을 구성하시오.

4. 자료기지보안에서 응용층에 있는 보안조종에 대한 실례들을 이미 고찰하였다.이 방법에서 제기되는 문제들은 어떤것들인가?

5. 대표값들이 자료항목에 비하여 민감도준위가 높은 자료기지를 생각하자. 자료항목에 접근하는 특권을 가진 사용자는 자료항목에 개별적으로 접근하는 방법으로 잠재적으로 집합체값을 계산할수 있다.
   이러한 공격을 어떻게 방어할수 있겠는가?

6. 표의 매행에서 따로따로 접근권한을 정의할수 있는 자료기지가 주어 져 있다. 접근조종은 조작체계의 보안기구들을 리용하게 될것이다.
   설계를 이와 같이 결정하면 자료기지대상들이 조작체계파일들에 기억되는 방법에 비하여 어떻게 되겠는가?(비교의 기준으로서 매개 파일의 행정자료가 100바이트인 조작체계와 1000만개의 기록으로 된 자료기지를 생각해 보시오.) 이것이 생활력 있는 설계결심으로 되겠는가?

제15장. 여러준위안전자료기지

이 장에서는 조종의 초점을 자료기지의 기입항목에로 완전히 옮긴다.

여러준위보안방책들은 자료에 대한 접근을 조종한다. 조작체계들에 있는 MLS와는 달리 기밀성문제와 안전성문제를 동시에 풀어야 한다.

잠복통로들은 피하여야 하지만 자료기지의 완정성속성들은 보존해 주어야 한다.

잠복통로들을 중지하기 위하여 보다 높은 준위에 있는 자료의 존재를 감추려고 한다면 관계형자료기지의 본질적인 완정성속성들을 보존하기 위해 다중구체례제시(polyinstantiation)에 의거하여야 한다.

보다 높은 준위에 있는 자료의 존재를 숨길 필요가 없다면 완정성을 보존하는것은 보다 쉽지만 잠복통로를 창조하지 않고 이러한 자료항목들을 자료기지에 삽입하는 방법을 찾아 내야 한다.

목적

- 자료기지체계의 배경에서 여러준위보안을 적용한다.
- 기밀성과 완정성의 요구사항사이의 잠재적인 모순을 분석한다.
- 이 모순을 해결하는 두가지 상반되는 방법들을 비교한다.
- 여러준위자료기지보안을 실현하는 두가지 상반되는 방법들을 비교한다.

제1절. 리론적기초

자료기지의 요소들이 너무 민감하여 가능한 가장 강한 보안대책만을 취하게 되는 세계를 상상해 보자. 사색이라는 하나의 수업을 거치면 위임접근조종이 다중준위안전자료기지에서 이 문제에 대한 해답으로 된다는것을 알수 있다.

위입접근조종을 관계형자료기지에 적응시키는데 상당한 지적인 노력이 소비되였다.

SeaView(SEcure dAta VIEW)[40]개발계획은 다중준위 안전관계형자료기지 관리체계(MLS-RDBMS)를 위한 원형을 배포하였다.

엄격히 보면 조작체계보안에서 쓰이는 개념들은 틀린 추상준위에서도 리용되지만 기밀성규칙들이 완정성을 위반하지 못하게 하는 실례들을 지적하고 있다.

이 장과 제16장에서 이 문제에 대한 실례들과 보안과 일관성사이의 실용적인 이률배반관계(trade off)의 실례들을 고찰하겠다.

대부분의 자료기지판매자들은 모두 다중준위자료기지보안을 지원하는 오렌지부크의 B1급에서 평가된 자기의 DBMS갱신판들을 가지고 있다. 자료기지판매자들은 지금도 일부 주문자들이 DBMS의 MAC특징을 리용하고 있으므로 이것을 리용할 결심을 내리기전에 모든 련관관계들을 품을 들여 고찰할것을 주문자들에게 경고하고 있다.

제2절. 관계형자료기지에서의 MAC

제4장 2절의 벨-라파둘라모형의 위임접근조종방책을 다시 상기하자. 간단히 고찰하기 위하여 주동체의 기정기밀취급허가와 현재의 기밀취급허가를 구별하지 않겠다. BLP모형에서 실체들은 다음과 같다.

- 주동체 모임S. 즉 자료기지체계의 사용자들.
- 객체모임. 즉 자료기지,기본관계들, 파생관계, 무이, 마당.
- 보안표식들의 부분순서화$(L, \leqq)$. 습관상 접근클라스라고 부른다.

함수 $fs:S \rightarrow L$은 접근클라스를 매개 주동체에 할당하며 함수 $f_O:O \rightarrow L$은 접근클라스를 매개 객체에 할당한다.

두가지 강제접근조종방책들은 정보가 접근클라스의 살창에 대하여 웃방향으로만 흐른다는것을 나타내고 있다.

규칙: ss- 속성(웃방향읽기 없음): 주동체 s는 s의 접근클라스가 객체 o의 접근클라스보다 웃준위에 있을 때 즉 $f_o(o) \leqq f_s(s)$일 때에만 객체 o를 관측할수 있다.

규칙: *- 속성(아래방향쓰기 없음): 객체 o의 접근클라스가 주동체 s의 접근클라스보다 웃준위에 있을 때 즉 $f_s(S) \leqq f_o(o)$일 때에만 주동체는 객체를 변경할수 있다.

이 방책들은 DBMS의 자료조작연산에 적용하면 직접적인 정보흐름문제를 취급할수 있다. 정보는 또한 잠복통로로도 흐를수 있다. 사용자가 사전에 그 요구가 비법적이며 반드시 실패한다는것을 모르고 있었다고 해도 그 접근요구를 거부하는것은 잠복통로를 형성한다.

1. 객체표시달기

SeaView에 따르면 가장 섬세한 준위에서 다중준위자료기지보안을 서술할수 있다. 자료기지의 매 항목은 자기의 표식을 받으며 이것은 자료요소, 무이, 관계 또는 자료기지일수 있다.

이것은 어느 한 무이가 각이한 보안표식들을 가진 요소들을 포함하고 있는 상당히 유연성이 있는 방책으로 된다. R가 n개의 속성을 가지는 다중준위관계라고 하자. 이때 R의 무이는 $(v_1, c_1, v_2, c_2, \ldots, v_n, c_n, t_c)$의 형태로 된다. 여기서 c_i는 i 번째 마당의 표식이며 tc는 무이의 표식이다. 사용자들은 보안표식을 볼수 없다.

보안표식들은 DBMS의 내부에 있는 정보로서 접근요구가 정당한가를 감시하는 참조감시기에서 리용된다.즉

- 자료기지표식: 사용자가 자료기지안에 있는 관계들을 지정할수 있는가를 결정하는데 쓰인다.
- 관계표식: 사용자가 자료기지안의 무이들을 지정할수 있는가를 결정하는데 쓰인다.
- 무이표식: 사용자가 무이안에 있는 모든 요소에 접근할수 있는가를 결정하는데 쓰인다.

- 요소표식: 사용자가 요소에 접근할수 있는가를 결정하는데 쓰인다.

앞장에서 보안방책이 다음과 같이 되여야 한다고 하였다.

- 완전성:자료기지의 모든 마당들이 보호된다.
- 일관성: 자료항목의 접근을 지배하는 규칙들은 모순이 없다.

자료항목들이 자기의 보안표식을 받았다면 보안표식의 할당은 끝난다. 따라서 완전성을 검사하는 방법은 단순하다.

2. 일관주소화

자료기지에서 여러가지 항목에 보안표식들을 할당하는 방법에 마땅히 주의를 돌려야 한다. 그렇지 않으면 표식이 쉽게 모순되는것으로 될수 있다. 일관성규칙들의 첫번째 모임에 대한 리유를 보기 위해 어떤 자료항목을 지정하기 위해 다음과 같은것들을 명기하여야 한다.

- 자료기지 D
- 자료기지 D안에 있는 관계 R
- 관계 R안에 있는 무이 r 의 1차열쇠
- 무이 r안에 있는 요소 r_i를 식별하는 속성 i

요소 r_i를 얻자면 다음의 관계가 성립되여야 한다.

$$f_o(D) \leqq f_o(R) \leqq f_o(r_i)$$

그렇지 않으면 보기 위해서 이름을 달아 준 요소에 대한 접근에 빗장이 걸릴수 있다. 우리의 경우에 무이 r에 접근하는 사용자는 자기의 모든 요소들에 접근한다. 이로부터 모든 속성 i 에 대하여

$$f_o(r_i) \leqq f_o(r)$$

을 요구한다.

요소의 표식달기는 관계형자료기지모형(제14장 2절 2)의 고유한 완정성규칙들을 일부러 다른 말로 바꾸어 표현하게 한다. 1차열쇠를 지정하면 다음과 같은 규칙을 얻게 된다.

규칙: 여러준위실체완정성: 기본관계에서 1차열쇠의 요소는 빈 원소로 될수 없다. 기본관계의 1차열쇠의 모든 요소들을 같은 접근클라스를 가진다. 기본관계에서 무이에 있는 다른 모든 자료값들의 접근클라스는 그 무이의 1차열쇠의 접근클라스를 지배한다.

외부열쇠는 다른 표에로의 련결이다.사용자가 이러한 련결을 보려고 한다면 그것을 따르도록 허가를 받아야 한다.

규칙: 여러준위참조완정성: 외부열쇠에 의하여 참조되는 무이는 반드시 존재하여야 한다. 외부열쇠의 접근클라스는 해당한 1차열쇠의 접근클라스를 지배한다.

3. 볼수 있는 자료

규칙의 다음모임은 여러가지 보안표식을 가지는 사용자들이 자료기지를 보게 하는 방법을 설명한다.

- 관계 R에 있는 무이들을 지정할수 있는 사용자 s에 대하여 $f_s(s) \geqq f_o(R)$ 가 되여야 한다.
- 자료원소 r_i 가 s에 보이자면 $f_s(s) \geqq f_o(r_i)$가 성립하여야 한다. $f_s(s) \not\supseteq f_o(r_i)$이고 r_i가 1차열쇠 r의 부분이라면 총체적인 무이는 볼수 없다. 어떤 다른 자료항목에 대하여 $f_s(s) \not\supseteq f_o(r_i)$이면 이 속성은 볼수 없으며 빈 원소값이 현시된다.
- 기본관계 R의 무이 r가 사용자 s에게 보여 진다면 $f_s(s') \geqq f_o(s)$로 되는 사용자 S'는 r의 값이 령 아닌 모든 속성에서 r와 일치하는 무이 r'를 볼수 있을것이다.

그림 15-11의 실례에서 이 규칙들을 설명하자. 호출클라스는 비밀에 속하지 않는 (U)와 비밀에 관계되는 (C)클라스가 있다. 자료요소의 보안표식들은 참조만 할수 있다. 앞에서도 이야기하였지만 보안표식을 사용자들한데 보여 줄수 없다. 접근표식 C를 가지는 주동체는 총체적인 표를 볼수 있지만(보안표식들이 아니라고 하여도) 접근클라스 U를 가지는 주동체는 그림 15-2의 목록에서처럼 비밀이 아닌 자료만을 볼수 있다.

4. 파생관계들

이제는 보임새, 순시상 또는 질문결과와 같은 파생관계들을 표식하기 위한 규칙에로 화제를 돌리자.

파생관계는 그의 정의를 평가함으로써 계산된다. 정보는 웃방향으로만 흐를수 있으므로 파생관계를 평가하는데 쓰이는 모든 자료의 접근클라스들은 결과의 접근클라스에 의하여 지배된다.

보임새에서 이 규칙은 다음과 같다.

> **규칙:** 보임새의 접근클라스는 보임새의 정의에서 리용되는 모든 관계의 접근클라스를 지배한다.

비행기 [FL_Class]		목적지 [DE_Class]		좌석 [Se_Class]		[Tuple_Class]
CA909	[C]	H.K.	[C]	7	[C]	[C]
AX301	[U]	K.L.	[U]	2	[U]	[U]
GR555	[U]	L.A.	[C]	11	[C]	[C]

그림 15-1. 1차열쇠 Flight(비행)를 가지는 관계 Bookings(예약)

비행기	목적지	좌석
AX301	K.L.	2
GR555	_	_

그림 15-2. 일반사용자에 접근할수 있는 그림 15-1로부터 비밀에 속하지 않은 자료의 목록

이 모형이 통계적자료기지보안과 상반되는 모형이라는데 주의를 돌리자. 여기서 집합체질문결과들은 개별자료항목에 비하여 민감하지 못할수 있다.

사용자 s가 파생관계를 평가할 때 파생관계에는 사용자접근클라스의 지배를 받는 접근클라스가 주어 지게 된다. 그렇지 않으면 사용자는 평가의 결과를 볼수 없게 될것이다.

사용자가 볼수 있는 파생관계를 구성하는 두가지 방법이 있다. DBMS는 먼저 사용자의 접근클라스에 독립인 파생관계를 평가하고 그다음 파생관계에 보안검사를 적용할수 있다. 또한 달리 DBMS는 이미 사용자의 접근클라스를 고찰한 다음 파생관계를 평가할수도 있다. 이때 결과에는 사용자가 볼수 있는 자료만이 있다. 어떤 순서를 택하든지 관계없이 그림 15-3에서 보여 준바와 같이 얻어 지는 결과는 같아야 한다. 보임새에 대한 이러한 교환요구를 형식화하면 다음의 규칙을 얻게 된다.

규칙: 주어 진 접근클라스에서 보임새의 실례는 그 접근클라스에서 보임새의 정의를 평가하는 결과와 같다.
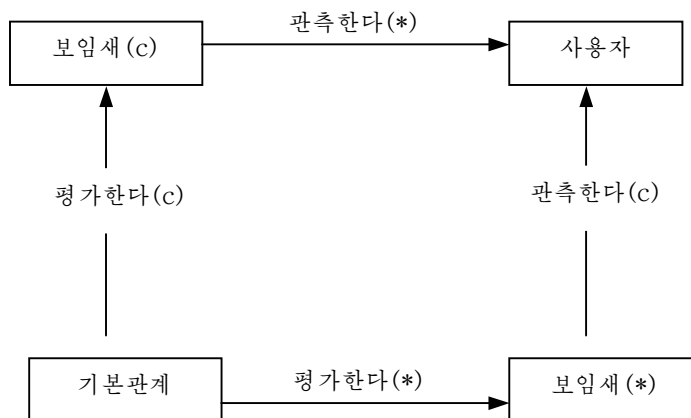
그림 15-3. 파생관계와 일치하는 평가

5. 위임접근조종

관계형자료기지에서 자료보안방책들은 보임새와 보임새우에서 접근권한으로 완전히 정의할수 있다. 보안방책들은 보임새에 대한 특권만을 허락하고 요소적인 SQL자료조작연산을 허락하지 않으므로 클라스-월슨보안모형과 로선이 완전히 같게 된다. 어떻게 하면 DAC방책을 위임접근조종에 맞게 작성할수 있는가?

만일 보임새가 《아래》사용자들에게 리용될수 있게 하려고 한다면 제15장 2절 4의 규칙들은 보임새에 의하여 고려되는 모든 자료항목들이 아래로 되여야 한다고 요구한다.

MAC에 따르면 이 자료항목들은 DAC의 방책의 의도를 명백히 고의적으로 위반하면서 《아래》사용자에게도 리용될수 있을것이다. 이 문제를 풀기 위하여 참조접근방식을 받아 들여 관계의 간접접근을 획득할수 있다. 이때 사용자에게는 보임새를 위한 특권과 구체적인 모든 (기본)관계우에서 참조방식을 위한 접근권한이 필요하다.

6. 기밀해제된 자료

위임접근조종은 아래방향에로의 쓰기를 금지한다. 이것은 정보의 민감도가 시간에 따라 떨어 지기때문에 관례적인 관찰에 맞지 않는 매우 제한적인 방책으로 된다. 실례로 회사의 결과들은 공개되기전에는 기밀성이 보장되여야 한다.

여러 나라들에서는 수십년이 지난 다음에야 정부가 기밀에 붙였던 파일들을 공개한다. 자료기지보안에서 정밀성을 요구하고 있으며 필요없이 제한하였던 기밀자료는 계속 보관하여서는 안된다.

따라서 아래에로의 통제된 쓰기를 진행하는 연산이 필요하다. 이 조작을 기밀해제라고 부른다. 자료의 기밀해제는 읽기클라스가 쓰기클라스를 엄밀히 지배하는 신용 받는 주동체에 의하여 실행될수 있다.

또한 파생관계를 평가함으로써 자료의 기밀해제를 실현할수도 있는데 이 경우 파생관계는 유도에 포함된 자료에 비하여 낮은 준위에서 분류된다. 지나친 분류를 피하기 위하여 DBMS는 보안방책이 허용되기만 하면 일정한 기간마다 시간에 관계되는 기밀해제 규칙을 적용하며 자료준위를 낮출수도 있다.

제3절. 다중구체례제시

여러준위보안 관계형자료기지모형에서 《아래》사용자는 《높은》 자료항목의 존재를 모르기때문에 아래사용자는 높은 값을 포함하고 있는 마당을 우연히 갱신하려고 시도할수 있다. 우리의 경우에 비밀문건을 다루지 않는 사용자가 례하면 《Dest=N.Y.》와 《Seats=0》과 같은 비밀이 아닌 값을 들여 보내여 비행기 GR555의 결여된 마당을 갱신하려고 할수도 있다.

이때 DBMS는 어떻게 반응하겠는가? 다음의 3가지 선택권이 있다.

- 갱신을 실행하는것을 거절한다: 그러나 이것은 이 위치에 《높은》 자료의 값이 들어 있다는 정보를 나타낸다.
- 낡은 값을 겹쳐 쓰기한다: 《높은》 자료의 값이 존재한다는 정보를 주지 않지만 이것을 파괴한다.
- 갱신을 실행하고 낡은 값을 보관한다: 이것을 다중구체례제시라고 부른다. 첫번째 선택은 보안의 리유로 그다지 사람들의 주의를 끌지 못한다.

위임접근조종을 리용하는 모든 수고를 아끼지 않는다면 이와 같은 눈에 띄우는 잠복통로를 받아 들여서는 안된다.

두번째 선택은 아래사용자들이 무의식적으로 높은 준위 자료를 파괴할수 있는 불우한 처지에 《높은》 사용자들이 있게 한다. 첫번째와 두번째 선택권들은 서로 배제적인 관계에 있으므로 다중구체례제시가 남는다. 우리의 경우에 예약(Bookings)관계는 그림 15-4에서 보여 준 표와 같이 갱신된다. 비밀에 속하는 준위에서 지워 진 사용자는 그림 15-5에서 보여 준 자료를 볼수 있다.

《다중구체례제시》라는 뜻은 동일한 1차열쇠에 여러개의 무이가 있을수 있다는것을 나타내고 있다. 다중구체례제시는 그림 15-5에서 명백히 알수 있다. 여기에는 1차열쇠 GR555에 대하여 두개의 기입항목이 있는데 이것은 1차열쇠의 정의에 명백히 위반되는것으로 된다. 보안을 강하게 하려던 나머지 얼핏 보기에도 알수 있는것처럼 관계형자료기지모형의 기초까지도 파괴하였다.

Flight[F1_Class]	Dest[De_Class]	Seats[Se_Class]	[Tuple_Class]
CA909 [C]	H.K. [C]	7 [C]	[C]
AX301 [U]	K.L. [U]	2 [U]	[U]
GR555 [U]	L.A. [C]	11 [C]	[C]
GR555 [U]	N.Y. [U]	0 [U]	[U]
GR555 [U]	N.Y. [U]	11 [C]	[C]
GR555 [U]	N.Y. [U]	0 [U]	[U]

그림 15-4. 그림 15-1에서 주어 진 표자료의 갱신판

보안의 요구를 관계형자료기지의 기본정의에 대한 요구와 일치시키기 위하여서는 무이에 있는 모든 마당의 접근클라스가 1차열쇠의 부분마당으로 된다고 선언해 주면 된다. 1차열쇠를 이렇게 확장하면 다시 무이의 유일한 식별이 보장된다.

관계의 무이를 주소화하기 위해서는 원래의 1차원열쇠외에 접근클라스의 벡토르를 지정하여야 한다. 비록 표의 표준형이 2차원구조이지만 다중준위표를 자리표계가 다음과 같은 3차원구조로 볼수 있다.

Flight	Dest	Seats
CA909	H.K.	7
AX301	K.L.	2
GR555	L.A.	11
GR555	N.Y.	0

그림 15-5. Confidential 사용자에 대한 자료호출

- (원본)1차열쇠
- 속성
- 접근클라스

문헌[40]에서 지적된 다음의 규칙은 다중구체례제시관계로부터 자료의 검색을 합리적으로 할수 있게 한다.

다중구체례제시완정성: 기본관계에 있는 두 무이가 동일한 1차열쇠를 가지며 어떤 속성에 대한 각각의 기입항목이 동일한 접근클라스를 가진다면 이 속성의 자료값들도 같다. 만일 기본관계의 두 무이가 동일한 1차열쇠를 가지며 각각의 기입항목이 서로 다른 접근클라스를 가지는 어떤 속성들이 있다면 그 속성들에 대한 값들은 서로 다를 수 있으며 이 값(호출클라스와)들의 임의의 조합은 다시 관계의 어떤 무이로 된다.

이 규칙의 첫 부분은 확장된 열쇠공간에서 어느 정도 준수된다.

다중준위관계를 나타내고 있는 3차원구조의 매점에는 많아서 하나의 값이 있을수 있다. 이 규칙의 둘째 부분은 다중구체례관계로부터 자료를 검색하기 위해서는 하나의 기초조작만이 필요하다는것을 확인하고 있다. 사용자가 확장된 1차열쇠에 대하여 Dest 와 Seats 의 값 Flight=GR555, Fl_Class=U, De_Class=U, Fl_Class=C을 요구한다고 가정하자.

대응하는 값 《Dest=N.Y》와《Seats=11》이 우연히 있지만 1차열쇠 GR555를 가지는 두개의 각이한 무이에서 존재한다. DBMS는 주어 진 1차열쇠가 이러한 요구를 처리하도록 하는 전체 무이를 검사하여야 한다.

다중구체례제시완정성규칙은 확장된 1차열쇠로 주소화될수 있는 값들의 전체 조합이 실제적으로 관계에 보관된다는것을 진술하고 있다. 우리의 실례에서 완전한 다중구체례제시관계는 그림 15-6과 같아 지게 된다. 질문결과는 이 표의 마지막 두번째 행에서 알수 있다. 의심할바없이 다중구체례제시완정성규칙은 갱신을 오히려 시끄럽게 만들며 다중구체례시화된 관계의 크기가 불어 나게 한다.

Flight[F1_Class]	Dest[De_Class]	Seats[Se_Class]		[Tuple_Class]
CA909 [C]	H.K. [C]	7	[C]	[C]
AX301 [U]	K.L. [U]	2	[U]	[U]
GR555 [U]	L.A. [C]	11	[C]	[C]
GR555 [U]	L.Y. [C]	0	[U]	[C]
GR555 [U]	N.Y. [U]	11	[C]	[C]
GR555 [U]	N.Y. [U]	0	[U]	[U]

그림 15-6. 열쇠 GR555의 모든 조합을 보여 주는 갱신된 자료표

제4절. 아래삽입

강제접근조종을 관계형자료기지모형에 첨가할 때 보안표식들을 DBMS의 내부정보로 취급하기로 결정하였다. 이렇게 하면 DBMS는 인증되지 않는 사용자들이 민감정보를 사용하지 못하게 할뿐아니라 지어는 민감자료의 존재를 은페시킨다. 한편으로는 이것이 더 많은 보안을 제공하지만 다른 한편으로는 잠복통로가 열리게 된다. 이 잠복통로를 닫기 위해서는 다중구체례제시에 의거하여 다중구체례제시를 관계형자료기지모형과 통합하는 어떤 방법을 찾지 않으면 안되였다. 관계형자료기지의 기초적인 완정성속성과의 치명적인 충돌을 협소하게나마 피할수 있었지만 이것은 시작에 불과하다.

제14장 2절 2에서 론의한 응용에 특정인 완정성제한조건을 주면 어떻게 되겠는가? MAC모형에 일치시키자면 모든 완정성제한은 접근클라스에 의하여 표식되여야 한다.

다음의 일관성규칙을 적용한다.

규칙: 완정성제한의 접근클라스는 제한이 적용되는 관계들의 접근클라스를 지배하여야 한다.

다중구체례제시가 도입된 리유가 바로 민감자료항목의 존재를 로출시키지 않으려는 목적에서였다는것을 상기해 보자. 완정성제한은 그것이 참조하는 자료항목보다 높은 준위에서 표식되는 경우에도 류사한 문제가 제기된다. 만일 낮은 준위의 주동체가 높은 준위의 제한을 받는 낮은 준위의 자료항목을 갱신하려고 한다면 DBMS는 주동체에 변경을 허락하여 잠재적으로 완정성제한을 위반하거나 일관성을 보존하기 위하여 높은 준위의 제한이 있다는것을 로출시키게 된다. 이때 이러한 함정에서 벗어 나게 하는 공학적인 방법은 없다.

자료기지는 외적사실들을 반영한다. 다중구체례제시된 자료기지는 외적사실들에 대하여 모호성을 만든다. 동일한 외적실체에 대하여 여러가지 기입항목들이 존재한다.

세계에 대한 각이하면서도 모순되는 견해를 주고 있는 자료기지를 어떻게 리용하겠는가? 표제기사가 있는 낮은 자료항목을 동일한 위치에 두면 불의에 나타나는 장애로부터 높은 준위 자료항목을 보호하기 위한 합법적인 위치가 얼마나 자주 존재하게 되겠는가? 더우기 표제기사가 높은 준위 자료항목들의 해당되는 측면들과 일치되지 않는 경우에만 일관성이 위반될 여지가 여전히 남아 있게 된다.

이 모든 문제들은 민감자료를 숨기도록 결정하였기때문에 생겨 난다. 자료항목들이나 제한조건들의 존재를 숨기지 않고 다중준위자료기지보안을 실현하여 그의 내용만을 보호해 줄수 있다. 이러한 정황에서는 존재를 로출시킨다고 하여 위법적인 정보의 흐름은 이루어 지지 않는다. 그것은 이 정보를 모든 주동체에 다 제공한다고 이미 결정하였기때문이다. 매개 자료항목과 관계(비록 무이는 아니라고 해도)에 보안표식을 다시 첨부하고 이 표식들을 그 관계를 보도록 확인된 모든 사용자에게 보이게 하겠다. 이 모형에서 일반사용자는 그림 15-1의 관계를 그림 15-7에서 주어 진것처럼 보게 된다.

다중구체례제시를 받아 들이지 않으면 안되게 하였던 요구에로 되돌아 가자. 일반사용자는 비밀이 아닌 값 《Dest=N.Y》와 《Seats=0》을 넣음으로써 비행기 GR555에 대한 빈 마당을 갱신하려고 시도한다. 이때 이 사용자는 기밀자료에 간섭하지 못하도록 작업을 끝낼것이 강요될수 있으며 관계는 변화되지 않고 이 관계의 1차열쇠를 변화시킬 필요도 없다.

비행기[Fl_Class]	목적지[De_Class]	좌석[Se_Class]
-　C	-　C	-　C
AX301　U	K.L.　U	2　U
GR555　U	-　C	-　C

그림 15-7. 비밀과 관련된 사용자에 접근할수 있는 자료의 갱신된 보임새

공개된 질문이 아직 하나 남아 있다. 비법정보흐름을 만들어 내지 않고도 자료가 어떻게 관계에 들어 가겠는가? 이 문제에 대한 답변을 하자면 우리의 실례에서 비정상을

수정하여야 한다. 일반사용자들은 1차열쇠를 모르고도 비밀에 속하는 무이가 있다는것을 안다. 참조완정성속성에 대한 우연적인 위반은 여전히 가능하다.

자료를 자료기지에 삽입하기 위하여서는 SWORD DBMS[161]가 제기한 아래삽입전략에 따라야 한다. 접근클라스《체계낮음》을 가지고 자료기지에서 새로운 항목을 창조할 과제를 가지고 있는 주동체《창조자》(creator)가 존재한다. 그외에는 어떤것도 그렇게 부를수 없다. 높은 준위자료항목을 창조하기 위하여 먼저 이 자료항목을 창조하고 거기에 어떤 위치표식정보를 기억시킨다. 따라서 이 자료항목이 있다는것을 누구나 다 알수 있다. 높은 준위의 주동체들은 아직 이 자료항목을 쓸수 없다. 다음에 창조자는 자료항목의 등급을 《높은》(high)으로 설정한다. 그 자료항목의 등급이 누구에게나 다 알려 진다. 그러면 높은 준위사용자들은 낮은 준위주동체에 의하여서는 접근될수 없는 자료항목에 비밀정보를 쓸수 있다.

접근될수 없는 낮은 자료를 포함하는 관계에 높은 준위 무이를 창조하기 위하여 관계를 서로 다른 부분으로 분할해 준다. 우리의 실례에서 창조자는 이름과 두개의 다음 관계에 대한 접근클라스 U_Bookings 와 C_Bookings 를 포함하는 공개된 표 All_bookings를 새롭게 구축한다. U_Bookings 관계는 1차열쇠가 공개되여 있는 Bookings 의 모든 기입항목을 포함하며 C_Bookings관계는 1차열쇠가 비밀로 되는 모든 기입항목을 포함하고 있다(그림 15-8).

일반사용자는 관계 C_Bookings 가 있다는것을 알지만 관계 U_Bookings에만 접근할수 있다.

마찬가지로 아래삽입방법을 응용에 고유한 완정성제한에 적용하면 제한의 내용을 숨길수 있으나 그의 존재는 감출수 없다. 높은 준위의 제한조건을 받는 낮은 준위자료항목을 갱신하려고 시도하는 낮은 준위주동체는 제한이 있다는 경고를 받고 갱신을 금지시킬수 있다.

SeaView방법에서는 보안표식들이 1차열쇠의 부분으로 되므로 열쇠공간이 실제적으로 확장되며 갱신 또는 선택과 같은 자료기지조작의 처리에서 변화가 생기게 된다.

All_Bookings

비행기	Bk_Class
U_Bookings	U
C_Bookings	C

U_Bookings

비행기[Fl_Class]		목적지[De_Class]		좌석[Se_Class]	
AX301	U	K.L.	U	2	U
GR555	U	L.A.	C	11	C

C_Bookings

비행기	Fl_Class	목적지	De_Class	좌석	Se_Class
CA909	C	H.K.	C	7	C

그림 15-8. 아래삽입접근전략의 실례

아래삽입에 의하면 다중구체례제시와 이것에 해당한 열쇠공간의 확장 그리고 관계들이 없이도 다중준위무이들을 창조할수 있다. 물론 아래삽입은 결함도 있다. 모든 자료항목들은 낮은 준위주동체에 의하여 창조되고 적당히 분류되여 있어야 한다.

모든 보안방책들이 이러한 분류절차와 다 일치하는것은 아니다. 이밖에도 주동체가 새로운 자료항목을 창조할 때마다 낮은 준위의 방조자를 필요로 하기때문에 높은 준위주동체의 작업은 무엇인가 지장을 받게 된다.

제5절. 실현에서의 문제

이 장에서는 여러준위보안 관계형자료기지체계에 관한 리론을 개략적으로 고찰하였다.이 리론을 더욱 전개하면 제16장에서 또 다른 측면을 보게 될것이다. 여기서는 MLS-RDBMS의 실천에서 제기되는 문제들을 론의한다. 이러한 체계를 실현하는데 두가지 서로 다른 방법들이 존재한다.

첫번째 선택은 자료기지체계를 여러준위보안조작체계의 맨 웃층에서 실행하는 봉사를 취급한다. MAC는 조작체계의 참조감시기에 의하여 시행된다. 이 전략에서 조작체계는 단일준위주동체들과 객체들을 처리하여야 한다.

- 매개 접근클라스에서 동작하는 개별적인 단일준위 DBMS처리가 있다.
- 다중준위관계들은 단일준위체계파일의 집합으로 기억된다.
- DBMS는 조작체계에 의하여 지원되는 접근클라스의 부분순서화를 리용하여야 한다.

조작의 이 방식은 B1체계들에 대하여 인증된 보안준위를 실현하는데 필요하다.

DBMS는 MAC 와 관계되는 한에서는 결코 TCB의 부분으로 되지 않는다. 조작의 이 방식은 또한 DBMS가 정상적으로 제공하는 여러가지 최량화방법들을 사전에 해볼수 있게 하므로 그다지 효과적인 방법이라고 볼수 없다. 더우기 표에로의 접근은 파일에 대한 다중접근조작으로 변환될수 있다.

이 방식에서 DBMS의 매개 구체적인 례는 자기의 준위나 보다 낮은 준위에 있는 자료만 볼수 있다. 따라서 보다 높은 준위 접근클라스에 있는 무이에 관하여 실체완정성의 특성을 위반하는것은 검출할수 없으므로 다중구체례제시가 자동적으로 발행된다.

두번째 선택에서 단일한 DBMS처리는 모든 접근클라스에 있는 자료에 접근하는 신용 받는 주동체처럼 동작한다. 이때 DBMS에는 MAC를 시행하는 참조감시기가 있다. DBMS 가 전체 자료기지를 알기때문에 실체완정성특성에 대한 위반들을 모두 검출할수 있다. 따라서 낮은 사용자가 우연적으로 높은 준위 자료항목을 갱신하려고 할 때 대처하는 방법을 결정할수 있다. DBMS 는

- 갱신을 진행하고 자료항목을 다중구체례제시할수 있다.
- 갱신을 부정하고 검열일지에 이 사건을 기록할수 있다.

두번째 경우에 DBMS는 그것을 즉석에서 차단하지 않고 잠복통로를 감시한다. 단일한 다중준위DBMS가 보다 효과적이지만 첫번째 선택보다 보증은 떨어 진다. 이때

DBMS는 TCB의 부분으로 되며 보안은 대규모적이며 매우 복잡한 쏘프트웨어에 기초하여 실현된다.이러한 해결책은 구체적인 안전한 다중준위조작체계를 요구하지 않는다. 특수한 사용자《Database》가 자료기지를 이루는 모든 파일들을 소유할수 있으므로 DAC 방책들은 다른 조작체계사용자들에 의하여 자료기지파일들이 호출되지 못하게 하는데는 충분하다. 검열기록에서 언급한것처럼 검열자료가 여러가지 보안준위들사이의 다른 하나의 잠재적인 정보통로로 된다는것을 잊지 말아야 한다. 따라서 검열기록들도 표식화되여야 한다. 검열선택권들은 체계에 기록될 사건들을 지적한다. 검열기록의 보안준위는 개개의 검열선택권을 정의한 사용자의 준위가 아니라 검열된 사건을 시동하는 조작을 진행한 사용자의 준위로 되여야 한다.

다중준위안전자료기지를 선택하는 경우에도 다중준위무이들이 실제로 필요한 유연성을 가져다 주는가 하는것과 그것으로 하여 골치거리가 생길수 있는가를 질문할수 있다.

자료기지방안을 합리적으로 설계하면 단일준위무이에서도 충분하다.관계에 서로 다른 보안준위의 무이들이 포함되여 있을수록 다중구체례제시로 되는 원인도 그만큼 많아지게 될것이다. 그러나 무이의 다중구체례제시는 자료항목의 다중구체례제시에 비하여 훨씬 관리하기 쉽다. 이것은 MLS-RDBMS 에서 찾아 볼수 있다.

이 장의 문헌안내

SeaView개발계획의 보안모형은 [40]과 [91]에서 구체적으로 서술되여 있으며 [25]에서도 취급되였다. 낮은 삽입전략은[161]에서도 제시되였다. 다중구체례제시와 완정성사이의 모순을 해결하는 가장 좋은 방법과 표제기사들의 리용에 관한 문제들은 [70,71]에서 론의되였다. 1980년대 말과 1990년대 초의 보안에 관한 학술토론회학회지에서 여러준위자료기지보안에 관한 많은 연구보문를 찾아 볼수 있다.

상업용으로 평가된 B1여러준위보안 자료기지제품들은 다음과 같다.

- INFORMIX-OnLine/Secure 4.1 과 5.0은 http://www.informix.com
- Trusted Oracle 7은 http://www.oracle.com
- Sybase SQL Server Version 11.0.6은 http://www.sybase.com

매개의 보증서들도 압축된 정보원천으로 되며 특히 보안을 실현하기 위하여 리용되는 체계등록정보도 자료기지보안에 대한 중요한 참고서라고 볼수 있다.

련습문제

1. SELECT, UPDATE, INSERT, DELETE, CREATE 등의 SQL 조작을 실현하는 위임접근조종을 정의하시오.

2. R가 n개의 속성 a1,...,an을 가지는 관계라고 하자. 관계 R 와 속성 a1,...,an 그리고 이 관계에 있는 모든 자료항목들은 표식화된다. 이 보안표식들에 따르는 일관성규칙을 표현하시오.

3. 관계 Accounts(구좌)는 1차열쇠가 Customer_Id 이고 속성이 Name(이름), Balance(잔고), Rating(리자률) 이다. 호출클라스는 3개 즉 비밀에 속하지 않는(U), 비밀과 관계되는(C), 비밀(S)가 있다. DBMS는 요소준위에서 다중구체례제시를 리용한다. 처음에 표는 비여 있다.다음과 같이 갱신을 하자.이때 매개 마당의 보안준위는 꺾쇠괄호안에 주었다.

   C01 [U] Kane [U] 15K [U] A [U]
   C15 [S] Hall [S] 300K [S] AA[C]
   C23 [U] Blake [U] 38K [C] B [C]
   C23 [U] Blake [U] 9K [U] A [U]

   - 두번째 갱신에서 일치하지 않는 리유는 무엇때문인가? 오유를 정정하시오.
   - 4번 갱신한 다음에 어느 기입항목들이 자료기지에 기억되는가?
   - U, C, S 준위의 사용자들은 4번 갱신한 다음 표를 읽을 때 무엇을 보게 되는가?

4. 아래삽입이 있는 DBMS는 우의 련습문제에서의 요구를 어떻게 처리하겠는가?

5. 표식화의 최소단위로 무이를 리용하는 관계형자료기지의 보안모형을 서술하시오.

6. 다중구체례제시는 관계형자료기지모형의 실체완정성규칙을 준수한다. 잠복통로를 창조하지 않고 여러가지 분류준위에서 자료항목을 계산하는 응용에 전용인 완정성규칙을 시행할수 있는 체계를 정의하시오.

7. 다중준위안전자료기지에서 여러가지 보안준위의 자료항목들은 자료기지질문의 선택평가기준에 맞을수 있다. 일부 자료항목들이 질문을 제기하는 사용자의 통과허가준위우에서 분류되였다면 DBMS는 정확한 대답을 줄수 없다. 가치 있는 답변을 주면서도 민감자료는 로출되지 않도록 질문을 수정할수 있는 서로 다른 방법들을 론의하시오.

8. 다중구체례제시를 실현하기 위하여서는 요소접근조작들이 자료기지에 적응되여 있어야 한다. 아래삽입방법을 실현하기 위하여 사용자들은 새로운 표를 창조할 때 개별적인 단계들을 거쳐야 한다. 두 경우에 체계의 어느 부분이 변경되여야 하는가? 어느 풀이가 높은 믿음성을 실현하는데 적합한가?

제16장. 병행조종과 여러준위보안

1장에서는 보안을 실현하는데 상당한 품이 요구된다는것을 강조한다. 앞장에서는 여러준위보안과 자료기지완정성사이의 모순점들을 고찰하였다.

이 장에서는 여러준위보안과 병행조종사이의 호상작용을 조사한다.

여러준위보안과 병행조종은 모두 확고한 리론적토대에 기초하고 있으므로 자료기지체계에서 보안과 리용가능성사이에 존재하는 리론적 및 실천적인 이률배반관계를 정확히 구분할수 있다.

목적

- 병행조종과 여러준위보안사이의 모순이 생길수 있는 원인을 리해한다.
- 병행조종을 간단히 소개한다.
- 직렬화가능성을 실현하면서 여러준위보안을 진행하는 두가지 병행조종방안들을 검토하고 매개의 약점들을 분석한다.
- 여러준위보안체계들의 비직렬성병행조종기구들을 분석한다.

제1절. 동기

자료기지에서 정보를 검색하는 주문자들은 일반적으로 응답시간이 뜬것을 싫어 한다. 따라서 DBMS는 두명이상의 사용자들이 동일한 자료항목을 동시에 접근하려고 하는 정황에 효률적으로 대처하지 않으면 안된다.이러한 경우에 어떻게 하여야 하겠는가? 한명의 사용자를 계속 기다리게 하겠는가? 이렇게 하면 자료기지의 리용률이 떨어 진다. 실례로 항공좌석예약체계에서 어떤 사람이 정기항공정보를 보고 있는 기간에 다른 사람은 기다렸다가 려행정보를 읽어야 한다.

한편 두 사람이 동일한 자료항목에 동시에 접근하게 하면 자료기지에 기억되여 있는 정보가 일치하지 않을수 있다. 실례로 동일한 은행구좌에 저금하는 두개의 업무를 고찰하자.

매개 업무는 현재의 차액잔고를 읽고 저금액수를 첨부하여 그 액수를 합하여 다시 자료기지에 써넣는다.

현재의 차액이 85원이라고 하자. 첫 업무는 100원을 구좌에 저금하고 두번째 업무는 25원을 저금한다고 하자. 이때 어떻게 되겠는가?

1. 첫번째 업무는 차액잔고를 읽고 85원을 검색한다.
2. 두번째 업무는 차액잔고를 읽고 85원을 검색한다.
3. 첫번째 업무는 새로운 차액잔고 85원+100원=185원을 계산하고 185원을 자료기지에 써넣는다.
4 두번째업무는 새로운 차액잔고 85원+25원=110원을 계산하고 110원을 자료기지에 써넣는다.

이때 자료기지에는 110원이 들어 있게 되며 100원은 없어 진다. 두개의 업무를 표현하는 지불전표와 같은 다른 자료가 더 있으면 좋을것이다. 이 업무들에 귀착되는 자료는 구좌의 잔고와 일치하지 않을것이다. 또한 구좌의 잔고는 그 소유자의 기대값과도 일치하지 않게 된다. 자료기지를 일치한 상태로 보관하는것은 필수적인 문제로 된다.

병행조종의 과제는 될수록 자료기지의 일관성을 위태롭게 하지 않으면서도 사용자들에게 많은 접근을 제공하는것이다. 다중준위안전병행조종알고리듬을 고찰하기 위하여 병행조종을 간단히 소개하려고 한다.

제2절. 병행조종

우리의 목적은 사용자들이 자기의 매개 프로그람이 더이상 분해할수 없는 단위로(원자적으로) 실행하는것처럼 생각하게 하는것이다. 다시말하면 마치도 동시에 실행되고 있는 다른 프로그람이 없는것처럼 보게 하는데 있다. 이와 같이 실행단위를 최소단위로 추상화하는것(원자적실행의 추상화)을 거래(transaction)라고 부른다. 여기서는 거래가 언제나 정확하다고 가정한다. 례하면 거래가 자료기지를 일치한 상태에 계속 보관한다고 가정한다.

거래는 자료기지기입항목들을 조작하기 위하여 자료기지조작들을 리용하는 프로그람이다. 사용자들이 제일 많이 쓰는 자료기지조작들은 읽기(read), 쓰기(write), 위임(commit), 포기(abort)들이다. 다음과 같은 의미로 쓰기로 하자.

- r_i: 거래 T_i 가 내보내는 자료항목 x에 대한 읽기조작
- $w_i[x]$: 거래 T_i 가 내보내는 자료항목 x 에 대한 쓰기조작
- C_i: 거래 T_i 가 성공적으로 끝날 때 내보내는 위임조작(완결조작)
- a_i 는 거래 T_i 가 포기하는 경우 내보내는 포기조작

식 $p_i[x]$ 또는 $q_i[x]$는 거래 T_i 에 의하여 발생되는 자료항목 x 에 대한 어떤 조작을 가리키는데 이것은 쓰기 또는 읽기조작으로 될수 있다.

DBMS가 매개 조작을 원자적으로 실행한다고 가정한다. 이와 같은 추상화준위에서 DBMS는 마치도 조작들을 순차적으로 실행하는것처럼 동작한다. 서로 다른 거래들에 속하는 조작들을 번갈아 실행함으로써 DBMS는 거래를 엇바꾸어 진행한다. 거래를 번갈아 진행하면 자료기지의 동시호출이 보다 효과적으로 되게 할수 있다. 방금 본것처럼 거래를 번갈아 진행하는것은 일관성이 위반되는 잠재적인 원천으로도 된다. 병행조종은 병렬로 동작하는 처리의 작용을 조장하며 공유자료에 접근하므로 서로 잠재적으로 간섭할수 있다.

정의: 병렬거래가 번갈아(엇끼우기) 실행할 때 이것이 직렬로 실행할 때와 자료기지에 주는 효과가 같다면 직렬가능하다고 한다.

직렬가능한 실행은 마치도 거래가 직렬로 실행되는것처럼 자료기지를 계속 동일한 상태에 남아 있게 하므로 정확히 실행된다. 매개 개별적인 거래가 일관성을 보존한다는 가정하에서는 거래호출의 임의의 직렬실행도 일관성을 보존할것이다.

이 론의는 BLP모형(제4장 2절)의 기본보안정리와 정확히 같다.

우의 실례로 되돌아 가면 두개의 예금업무를 직렬로 실행할 때 첫번째 업무(거래)는 85원을 읽고 100원을 더하며 다시 185원을 쓴다. 그러면 두번째 업무에서는 185원을 읽고 25원을 더하여 210원을 쓴다.

직렬화가능성을 실현하자면 충돌하는 조작들에 의해 발생하는 문제들을 고찰하여야 한다.

정의: 두개의 조작들이 서로 다른 거래에 속하고 같은 자료항목에 조작을 진행하며 적어도 하나의 조작은 쓰기이면 이 두개의 조작들을 충돌이라고 부른다.

우리의 실례에서 두개의 거래들은 다른 거래가 발생하는 《읽기》 및 《쓰기》와 충돌하면서 자료항목《차액잔고》(balance)에 써넣었다.

1. 적극적 및 보수적일정작성프로그람

병행조종을 처리하는 DBMS부분은 거래를 처리하고 자기의 조작을 일정작성프로그람에 넘겨 주는 거래관리자(TM)를 포함하고 있다. 일정작성프로그람은 조작을 실행하겠는가 그리고 언제 실행하겠는가를 자료관리자(DM)에게 물어 보고 결정한다. 일정작성프로그람이 TM으로부터 조작을 접수할 때 다음의 3가지 선택항목을 가진다.

1. 즉시에 조작의 일정을 작성한다.
2. 조작을 지연시켰다가 후에 다시 고려한다.
3. 조작을 기각시킨다.

적극적일정작성프로그람(aggressire scheduler)은 이후에 받게 되는 조작들의 순서를 다시 정할 기회를 앞질러 나가면서 즉시에 조작들의 일정을 작성함으로써 지연을 피하려고 시도한다.

적극적일정작성프로그람은 모든 능동적인 거래를 직렬가능한 실행으로 끝낼 가망이 없는 정황에 빠질수 있다(거래가 시동은 되였지만 끝나지 않거나 포기되지 않는다면 능동이라고 말한다).

이 시점에서 적극적일정작성프로그람은 하나 또는 그이상의 거래를 포기하고 재시동하여야 한다(roll-back).

적극적일정작성프로그람은 병행조종을 최량적으로 실현한다. 최량적인 병행조종은 충돌이 드물게 발생하여 재시동이 거의 진행되지 않는 경우에 적합하다.

보수적일정작성프로그람(conservative schedaler)은 이후에 접수한 조작들의 순서를 다시 정하는데 보다 많은 여유를 얻으려고 조작들을 지연시키려고 한다.

보수적일정작성프로그람은 직렬가능한 실행을 산생시키는 조작들을 기각시켜야 하는 정황에서는 적합하다고 볼수 없다.

극단한 경우에 보수적일정작성프로그람은 거래를 직렬로 처리한다.

적극적일정작성프로그람과 보수적일정작성프로그람사이에는 명백히 이률배반관계가 있다.

- 적극적일정작성프로그람은 조작들이 지연되는것을 피하므로 이후에 그에 의한 위험성은 제거된다.
- 보수적작성프로그람은 조작들을 심사숙고하여 지연시킴으로써 조작들이 기각되는것을 피한다.

지나친 제한을 피하기 위하여 보수적일정작성프로그람은 아직 수신되지 않은 조작들을 될수록 정확하게 앞질러 처리하여야 한다. 매개 거래에서 필요한 기본정보는 자료항목의 모임에 대하여 읽기와 쓰기를 하는 readset와 writeset 이다. 아주 보수적이라고 볼수 있는 일정작성프로그람을 구축하는데서 장애물로 되는것은 주어 진 하나의 프로그람을 각이하게 실행시키면 거래가 자료항목의 각이한 모임에 접근하게 될수 있다는것이다.

따라서 거래는 읽기 또는 쓰기할수 있는 전체 자료의 모임을 미리 선언해 주어야 한다. 이때 흔히 거래가 자기의 readset 와 writeset 를 과장하게 한다. 거래가 고수준질문언어를 리용하여 DBMS와 작용하는 경우에도 우와 같은 문제가 발생할수 있다.

거래가 readset 와 writeset 를 하는 모임을 실제보다 크게 나타낸다면 일정작성프로그람은 앞으로 절대로 발생되지 않는 기타 조작들까지도 미리 예견하여 조작들을 지연시킬수 있기때문에 필연적이라기보다도 보수적이라고 말하는것이 더 좋을것이다.

2. 2-상잠금

2-상잠금은 오늘 DBMS 제품들에서 쓰이고 있는 가장 일반적인 병행조종기구이다. 잠금방법은 공유자료에 대한 접근을 동기화하는데 쓰이는 가장 일반적인 수단으로 되고 있다. 매개 자료항목은 자기와 관계되는 자물쇠를 가지고 있다.

$rl_i[x]$가 자료항목 x에 대한 읽기자물쇠, $wl_i[x]$가 거래호출 T_i에 의하여 얻어 진 x에 대한 쓰기자물쇠를 나타낸다고 하자. 이때 $ru_i[x]$와 $wu_i[x]$은 T_i가 x에 대한 읽기 또는 쓰기자물쇠를 해제하는 조작을 나타낸다. 마찬가지로 $pl_i[x]$와 $ql_i[x]$는 $p_i[x]$와 $q_i[x]$의 조작에 필요한 x에 대한 자물쇠를 나타낸다. 대응하는 자물쇠열기조작들은 $pu_i[x]$와 $qu_i[x]$이다.

규칙: 두개의 자물쇠가 동일한 자료항목에 걸려 있고 서로 다른 거래에 의하여 발생되며 그들중 적어도 하나는 쓰기자물쇠라면 두 자물쇠는 충돌한다.

보다 형식적으로 말하여 $x = y$, $i \neq j$ 이고 p와 q의 조작들이 상반되는 형을 가진다고 하면 두 자물쇠 $pl_i[x]$와 $ql_j[x]$는 충돌한다.

일정작성프로그람은 확고히 서로 다른 거래에 허락되는 자물쇠중에서 충돌하는 자물쇠가 없게 해줌으로써 자료항목에 대한 접근이 충돌하는것을 막아야 한다. 기본2-상잠금일정작성프로그람(2*PL*)은 아래의 규칙에 따라 거래가 자기의 자물쇠를 얻고 해제하는 시간을 조종하는 방법으로 자물쇠를 관리한다.

규칙 1: 일정작성프로그람이 *TM*으로부터 $p_i[x]$조작을 받으면 $pl_i[x]$는 이미 설정된 어떤 자물쇠 $ql_j[x]$와 충돌하는가를 검사한다. 만일 충돌한다고 하면 $p_i[x]$를 지연시켜 필요한 자물쇠를 설정할수 있을 때까지 T_i를 기다리게 한다. 충돌하지 않는 경우 일정작성프로그람은 $pl_i[x]$를 설정하고 $p_i[x]$을 *DM*에 보낸다.

규칙 2: 일정작성프로그람이 일단 T_i에 자물쇠 $pl_i[x]$를 설정하기만 하면 적어도 그 자물쇠에 해당한 조작 $p_i[x]$를 처리하였다는것을 *DM*이 인정하기전에는 그 자물쇠를 절대로 해제할수 없다.

규칙 3: 일정작성프로그람이 거래에 쓰이는 자물쇠를 해제하였다면 그 거래에서는 그밖의 자물쇠를 뒤이어 얻을수 없다.

규칙 1은 두개의 거래가 충돌방식으로 자료항목을 동시에 호출하지 못하도록 한다.

규칙 2는 규칙 1에 일정작성프로그람이 허용하는 순서로 DM이 자료항목에 대하여 조작을 실행한다는 담보를 보충적으로 주고 있다.

2-위상규칙이라고도 하는 규칙 3에서는 매개 거래의 실행을 아래의 두개 위상으로 구분한다.

- 증가위상, 거래가 자물쇠들을 얻는 기간.
- 축소위상, 거래가 자물쇠들을 해제하는 기간.

2PL은 병렬적인 거래들에 속하는 충돌조작들에 이 거래를 직렬실행순서와 같게 순서작성한다.

이와 같은 견해는 다음과 같은 정리의 기초로 된다.

정리: 2-상잠금은 직렬가능한 형태로 거래의 교차적실행을 쉽게 한다.

그러나 기초적인 2-상잠금은 영구대기(deadlock)를 극복하지 못하므로 이것을 해결하기 위한 기구가 보충적으로 필요하다. 이것은 사실상 우리의 실례에서 틀림없이 일어나게 될것이다. 거래호출 T_1와 T_2은 《차액잔고》에 관하여 읽기자물쇠를 얻는다. 그이후부터는 두개의 거래호출이 모두 차단되게 된다. 그것은 두개의 거래가 다른 거래에 의하여 점유되는 읽기자물쇠와 충돌하는 쓰기자물쇠를 필요로 하고 있기때문이다.

3. 다중판본시간도장순서화

다중판본시간도장순서화(multi-version timestamp ordering)는 다른 하나의 병행조종기구이다. 다중판본자료기지는 매개 자료항목의 다중판본을 기억할수 있다. DBMS는 이 사실을 사용자가 모르게 숨긴다.

거래는 개별적판본을 기준으로 작성되지 않았으므로 DBMS는 거래를 자료항목의 해당한 판본으로 넘겨 실행한다. DBMS는 거래가 접근하는 자료항목의 판본을 선정할 때 일관성조건을 고려할수 있다. 따라서 동일한 자료에로 접근하는 거래는 그 자료항목의 여러가지 판본에 대처할수 있으므로 불일치한 상태로 될 가능성도 첨부된다.

다중판본자료기지에서 거래를 교차식으로 진행하기 위하여서는 다음과 같은 성질을 가져야 한다.

정의: 다중판본자료기지우에서 거래모임의 교차식실행이 단일판본자료항목으로 된 자료기지에서 거래의 어떤 직렬실행과 동등하다면 1회복사직렬가능하다고 한다.

다중판본시간도장순서화방법(MVTO)은 다중판본자료기지에 적합한 병행조종알고리듬이다. 자료항목들은 여러개의 판본을 가진다.

자료항목 X 의 i 번째 판본을 $X.i$ 로 표기하자. MVTO 일정작성프로그람들은 아래의 시간도장을 리용하는데 시간도장을 할당하는 방법에서 일정작성프로그람들은 차이가 있다.

- 매개 거래에 고유한 시작시간이 주어 진다. 두개의 거래에 동일한 시작시간을 할당할수 없다. 시작시간은 체계박자(system clock) 또는 계수기(counter)로부터 유도할수 있으며 보통 어떤 거래의 첫번째 조작이 순서 정해 진 시간을 가리키지만 시작시간이 반드시 그 거래가 실제적으로 시작된 시간은 아니다.

- 모든 쓰기조작은 쓰기시간도장이 할당된 자료항목에 대하여 새로운 판본을 창조한다. 쓰기시간도장은 쓰기조작을 진행한 실제적인 시간과 반드시 일치하지 않아도 된다.
- 매개 읽기조작에는 읽기시점(read point)이 할당되여 있다. 읽기요구는 읽기시점보다 앞에 있는 가장 큰 쓰기시간도장을 가지는 자료의 판본으로 넘겨 진다.
- 자료항목의 매개 판본은 읽기시간도장을 가진다. 읽기조작은 읽어 지는 자료항목의 판본에 대한 읽기시간도장을 갱신한다. 새로운 읽기시간도장은 현재읽기시간도장 또는 조작의 읽기시점중에서 최신의것이 된다. 여기에서도 읽기시간도장이 반드시 마지막으로 판본을 읽은 시간으로 되여야 한다는 법은 없다.

정의에 따르면 자료항목의 어떤 판본에 대한 쓰기시간도장은 언제나 읽기시간도장보다 앞선다.

거래의 시작시간을 모든 읽기조작의 읽기시점과 모든 쓰기조작의 쓰기시점으로 리용하는 MVTO순서일정알고리듬을 고찰하자.

은행구좌번호의 실례를 여러 판본자료기지에서 표현하면 조작의 렬은 다음과 같이 쓸수 있다.

조작	start1	r1[b.1]	start2	r2[b.1]	w1[b.2]	w2[b.3]
박자표식	1	2	3	4	5	6

차액잔고에 대한 판본 $b.1$의 쓰기시간도장이 0 또는 어떤 초기값이라고 가정하자. $b.1$의 읽기시간도장은 첫번째 거래의 시작시간으로써 먼저 1로 갱신되였다가 다음에 3으로 갱신된다.

판본 $b.2$ 의 쓰기시간도장은 1이며 그 값은 이 실례에서 185원이다.

판본 $b.3$ 에 대한 쓰기시간도장은 3이며 그 값은 110원이다.

이 실례의 은행업무처리와 같이 첫번째 거래가 자료항목을 읽고 이것이 끝나기전에 두번째 거래가 그 자료항목에 대한 새로운 판본을 쓰려고 하는 경우에는 문제가 생긴다. 따라서 MVTO 일정작성프로그람은 만일 새로운 쓰기시간도장이 쓰려고 시도하는 자료항목의 판본의 쓰기와 읽기시간도장사이에 있다면 쓰기조작을 기각시켜야 하며 쓰기요청을 내보내고 있는 거래를 포기하여야 한다. 보다 형식적으로 말하면 자료항목 x에 대한 쓰기조작은 새로운 판본 $x.$new을 창조하기때문에 만일 다음의 관계식에 성립하는 판본 $x.i$ 가 존재한다면 기각되여야 한다.

$$\text{write timestamp}(x.i) < \text{write timestamp}(x.new) < \text{read timestamp}(x.i)$$

시간도장을 이와 같이 설정해 주는 방책과 방법에 의해 MVTO순서작성프로그람은 1회복사직렬화가능성을 담보한다.

우리의 실례에서 첫번째 거래가 차액잔고를 새롭게 쓰려고 시도하는 경우에는 이것을 포기할것이다. 그것은

$$\text{write timestamp}(b.1)=0 < \text{write timestamp}(b.2)=1 < \text{read timestamp}(b.1)=3$$

이기때문이다.

제3절. MLS병행조종

병행조종은 공유된 자료에 대한 접근을 동기화한다. 동기화는 여러가지 형태의 통신을 필요로 한다. 자료항목이 현재 입력될수 없기때문에 거래가 지금 진행될수 없다는것을 최소한 거래에 알려 주어야 한다. 이러한 통보문에는 자료항목을 차단하는 거래로부터 자료항목을 기다리는 거래까지의 정보흐름이 포함되여 있다. 여러준위보안에서는 비법적인 정보흐름에 대하여 불안한감을 가지고 있다. 명백하게 말하여 병행조종과 여러준위보안은 서로 다른 방향에서 론의되는 문제들이다.두가지 측면을 고려하여 다중준위안전병행조종알고리듬을 찾아 낼수 있겠는가?

1. 일반적관찰

표준MLS모형에서 모든 주동체들과 객체들은 보안표식을 가지고 있다. 자료기지의 거래와 조작들의 접근클라스(보안표식)들은 다음과 같이 결정될수 있다.

- 사용자가 체계에 가입하면 사용자가 가입한 통과허가를 가지고 동작하는 처리가 시작된다.
- 이 처리가 거래를 시작하면 거래는 처리의 접근클라스를 계승한다.
- 이 거래의 어떤 조작을 내보내면 조작은 거래의 접근클라스를 계승한다.

그러면 제15장 5절로 되돌아 가자. 여러준위보안이 가장 강한 상태에 있자면 오직 하나의 단일준위주동체가 있어야 한다. 매개 주동체가 자료기지의 접근을 필요로 하고 있을 때 조작체계는 그 주동체의 접근클라스판본을 가지는 DBMS을 구체례제시화하여야 한다. 따라서 단일한 DBMS가 아니라 서로 다른 보안준위에서 개별적인 DBMS들이 존재하게 된다.

병행조종기구들은 자료기지의 대역적인 특성을 만족하여야 한다.

어떤 특성이 전체 자료기지에 대하여 준수되여야 한다면 대역적이라고 한다.

어떤 특성이 자료기지의 부분모임에서만 실례로 하나의 개별적인 보안준위에서 준수되여야 한다면 국부적이라고 한다.
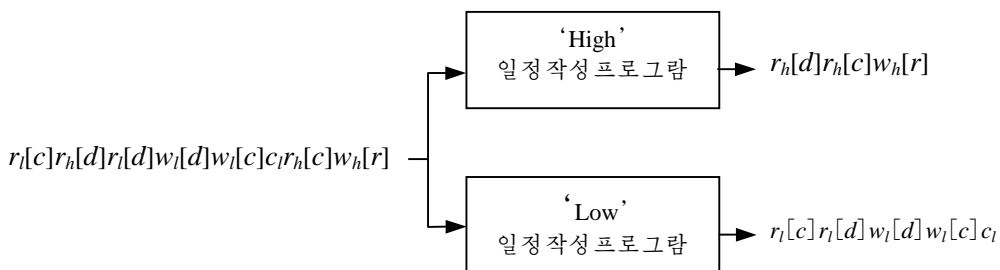
그림 16-1. MLS환경에서 일정작성

만일 여러가지 접근클라스에 있는 단일준위주동체들이 서로 작용하지 못하도록 보안방책을 작성한다면 비록 그것들이 대역적인 특성을 만족시키기 위하여 서로 협동, 협력하여 작용한다고 하여도 대역적특성을 만족하기가 곤난하거나 불가능할수도 있다. 보안

의 요구가 없다면 단일한 일정작성프로그람이 2-상잠금방법을 리용하여도 직렬화가능성을 담보할수 있다.

그러나 여기서 고찰하는 보안모형에서는 일정작성프로그람은 보안표식을 배정 받는 주동체이다.

- 모든 주동체들이 단일준위에 있다면 대역적인 일정작성프로그람은 있을수 없다.
- 매개 접근클라스에서 자기의 클라스에 대한 거래만 작성하는 단일준위일정작성프로그람은 반드시 하나여야 한다.
- 단일준위일정작성프로그람들은 총체적으로 대역적인 직렬화가능성을 담보하여야 한다.

일정작성에 의해서가 아니라 TCB에 의하여 보안이 담보된다고 볼수 있는것은 보안에서 취급할 문제가 아니다. 앞에서 론의한 전략을 다시 검토하고 새로운 요구에 적응시켜야 한다.

BLP보안방책들은 충돌을 취급할수 있는 방도로 되기때문에 다음과 같은 의의를 가진다.

- 쓰기낮음/읽기높음: 높은 준위 일정작성프로그람은 낮은 일정작성프로그람들의 동작을 보고 이 동작들에 반작용할수 있다. 낮은 일정작성프로그람은 높은 준위 일정작성프로그람의 동작을 보게 되여 있지 않으므로 높은 준위 거래호출에 무관계하게 결정할수 있다.
- 쓰기높음/읽기낮음: 이것은 BLP 보안방책들에 의하여 금지된다. 여기서 보안은 충돌의 수를 제한하므로 실제적으로 일정작성프로그람에게 편리하다.
- 동일한 준위에서 충돌하고 있는 접근조작들은 바로 그 준위에 있는 단일준위일정작성프로그람에 의하여 처리될수 있다.

앞에서 우리는 잠금과 시간도장의 두가지 기본적인 일정작성방법들을 언급하였다. 잠금규약에서 다음의 정황을 생각해 보자. 준위 h(높음)에서 거래 T_h가 자료항목 X를 읽고 그다음 준위 l (낮음)에서 거래 T_l, $(l < h)$이 동일한 자료항목 X에 쓰기를 한다.

이 충돌은 잠금에 의하여 해결될수 있는가

T_h에 의하여 설정된 읽기자물쇠의 보안준위에 따라 여러가지로 말할수 있다.

낮은 준위로 자물쇠가 분류되였다면 낮은 일정작성프로그람은 자물쇠를 보게 될것이며 T_l 에 의하여 발생되는 쓰기조작을 차단할수 있다.

그러나 높은 준위 일정작성프로그람이 BLP규칙들을 위반하면서 낮은 자물쇠를 쓰기할 때 쓰기낮음이 이루어 진다. 따라서 T_h에 의하여 설정된 읽기자물쇠는 높은 준위로 분류되여야 한다. 읽기자물쇠는 낮은 일정작성프로그람에게 보여 질수 없으므로 충돌을 해결할수 없다.

MVTO일정작성프로그람에서 높은 준위 거래 T_h가 낮은 거래 T_l보다 후에 시작하지만 T_l이 자료항목 x(l 준위에 있는)를 쓰기하기전에 x를 읽어 내는 경우를 생각해 보자. 이때 충돌이 발생한다. 그것은 x에 대한 현재판본의 읽기시간도장이 T_h의 시작시간으로

되겠지만 새로운 판본의 쓰기시간도장은 비록 후에 창조되였다고 하여도 읽기시간도장보다 앞선 T_l의 시작시간으로 되기때문이다.

이 충돌을 MVTO로 해결할수 있는가

이 경우에도 높은 준위일정작성프로그람이 쓴 시간도장들은 높은 준위로 분류하여야 하므로 낮은 준위일정작성프로그람은 높은 준위의 거래를 여전히 모르고 있다는 문제에 귀착된다.

높은 준위일정작성프로그람만이 충돌을 해결할수 있다. 높은 준위일정작성프로그람은 일관성을 담보하는 다음의 두가지 선택권을 가지고 있다.

- 낮은 준위 거래와 하나도 충돌하지 않을 때까지 높은 준위 거래를 지연시킨다(최량병행조종). 또는
- 높은 준위 거래에 충돌을 발생시킬수 있는 가능성이 있는 자료항목의 낡은 판본에 높은 준위 거래의 접근권한을 부여한다(여러판본시간도장순서화).

첫번째 경우에 높은 준위 거래는 기다리는데 싫증날수 있다.
두번째 경우에 높은 준위 거래호출은 이미 낡아 쓸모가 적은 정보에 작용할수 있다.
이 두가지 해결책을 좀더 구체적으로 고찰하자.

2. 최량적인 MLS병행조종

최량적인 병행조종은 매개 자료항목의 하나의 복사본만이 존재하는 단일판본자료기지에서 실현된다.

이때 거래는 자기의 실행을 계속할수 있고 따라서 결과는 다른 처리에로 넘어 가기(commit)전에 유효하게 된다. 어떤 충돌이 이 단계에서 발견되였다면 검사회복(roll-back)이 발생하여 거래가 다시 시동된다.

거래가 《충돌이 일어 나지 못한다는 기대를 가지고》 실행될수 있으므로
이 전략은 최량적인 병행조종의 한가지 실례로 된다고 볼수 있다.

거래는 3개의 위상에서 실행되며 보안준위가 거래와 같은 일정작성프로그람에 의하여 조종된다.

- **읽기위상:** 주동체는 자기가 요구한 자료항목의 비공개복사를 얻고 이것을 갱신한다. 일정작성프로그람은 이 위상의 시작점을 start-read timestamp 로 표시한다.
- **유효위상:** 일정작성프로그람은 transaction number로 되는 start-validation timestamp를 설정한다. 일정작성프로그람은 충돌을 검사하고 충돌이 검출되였다면 회복처리가 발생하며 거래는 재시동된다.
- **쓰기위상(비여 있을수도 있다.):** 충돌이 없다면 거래는 다음의 처리에로 넘어(commit) 가며 자료항목의 비공개복사는 공개복사로 된다.

높은 준위의 거래가 유효할 때 다음의 세가지 조건이 동시에 만족시키는 경우에는 충돌이 검출된다.

1. 보다 낮은 준위의 거래에 의하여 충돌을 일으키는 자료항목이 존재한다.

2. 보다 낮은 준위의 모든 거래가 완료 또는 포기되기전에 높은 준위의 거래가 자기의 읽기위상을 시작하였다. 결과 높은 준위읽기조작들은 자료기지의 서로 다른(일치한) 상태들에 접근하였을련지도 모른다.
3. 보다 낮은 준위의 어떤 거래가 자기의 유효위상에 들어 간 다음 높은 준위거래가 자기의 유효위상에 들어 간다. 그렇지 않은 경우에는 그의 읽기조작들이 자료기지의 동일한(일치한) 상태에 대하여 실행되였다고 담보된다.

최량적인 단일준위일정작성프로그람은 보다 낮은 준위의 거래와의 충돌을 검출하고 필요하다면 자기 준위의 거래를 포기할수 있다.

이때 매개 접근클라스에 대하여 시간도장(거래의 고유한 번호)을 발생하기 위한 체계낮음(론리)박자가 필요하게 된다. 박자수당 오직 하나의 번호만이 발생한다.

단일준위일정작성프로그람은 시작읽기시간도장, 거래번호, 자기 준위의 거래에 대한 모임읽기를 보관하며 다음과 같은것을 읽어야 한다.

- 보다 낮은 준위에 있는 거래의 writeset지령
- 보다 낮은 준위의 거래에 대한 유효위상의 시작시간, 즉 거래번호
- complete-write지령.보다 낮은 준위의 거래에 대한 쓰기완료(포기)시간

일정작성알고리듬은 접근클라스 h에 있는 거래 T_h를 처리할것인가 말겠는가를 다음과 같이 결정한다[60].

T_h 가 현재 유효단계에 있다고 한다.
IF 접근클라스 l(l < h) 에 거래 T_n 가 존재하면
거래-번호(T_l) < 거래-번호(T_h), 그리고
start-read(T_h) < complete-write(T_l), 그리고
writeset(T_l) ∩ readset(T_h)≠0
THEN roll back and restart T_h
ELSE commit T_h

MLS 순서작성과정을 설명하기 위하여 다른 실례를 들자.

《low》로 표식화된 두개의 은행구좌번호 c(류통), d(예금)를 가진 거래자를 고찰하자. 은행에는 또한 《high》로 표식화된 상환능력성능평가지표가 보관되여 있다. 이 평가지표는 c 와 d 의 차액잔고가 1000원보다 많으면 A가 되여야 하며 그렇지 않으면 B로 된다. 높은 준위사용자들만이 상환능력성능평가지표를 갱신할수 있다. 처음에 구좌번호가 c:300, d:800, r:A 의 값을 가진다고 하자. 높은 준위거래 T_h가 상환능력성능평가지표를 검사하고 있는 동안 낮은 준위거래 T_l은 c로부터 d로 200원을 전송한다. 다음과 같은 사건들의 렬을 생각해 보자.

조작:	$r_l[c]$	$r_h[d]$	$r_l[d]$	$w_l[d]$	$w_l[c]$	$startvall$	c_l	$r_h[c]$	$w_h[r]$	$startval_h$
박자수:	1	2	3	4	5	6	7	8	9	10

거래 T_l은 c:300, d:800을 읽고 구좌번호를 c:100, d:1000으로 갱신한다. 거래 T_h는 T_l에 의하여 갱신되기전에 d:800을 읽고 다음에 T_l에 의하여 갱신된후에 c:100을 읽는다. T_h는 불일치한 평가지표 r:B를 쓸것이므로 포기되여야 한다. 3가지 충돌조건들이 다 들어 맞는다.

이 실례에서 검사결과는 다음과 같다.

- transaction-number(T_l)=6 < transaction-number(T_h)=10;
- start-read(T_h)=2 < complete-write(T_h)=7;
- writeset(T_l) ∩ readset(T_h)={c}

따라서 T_h는 포기되여야 한다.

3. 단일준위일정작성프로그람이 있는 MVTO

단일준위일정작성프로그람을 가지는 여러판본시간도장은 이후에 낮은 준위의 쓰기조작이 반드시 높은 준위 거래를 유효로 할수 있다는것을 사전에 담보함으로써 충돌을 피한다. 이것을 실현하기 위하여 높은 준위의 거래가 너무 낡아 보다 낮은 준위의 거래에 의하여 절대로 접근되지 않는 자료항목의 판본을 강제로 읽게 한다.

거래의 시작시간을 조종함으로써 거래가 읽는 자료항목의 판본을 조종할수 있다. 높은 준위 거래에 보다 낮은 준위에 있는 능동적인 어떤 거래의 시작시간보다 앞에 있는 지난 시기의 어떤 시작시간을 줌으로써 능동적인 낮은 준위 거래의 쓰기조작과 절대로 충돌이 일어 날수 없다는것을 확신할수 있다.

이 일정작성알고리듬에서 거래의 시작시간도장은 실제의 시작시간시간도장과 같지 않을수도 있다. 이때 시작시간, 읽기시간 또는 쓰기시간도장은 사건을 일으킨 실제적인 시간들에 대응하지 않는다.

매개 접근클라스에서 단일준위일정작성프로그람은 자기가 있는 준위의 거래에 시간도장을 할당한다. 거래가 시작하면 지연된 시작시간을 수신하며 이 시간은 보다 낮은 준위의 모든 능동거래보다는 앞에 있고 자기가 있는 준위에서 시작한 모든 거래보다는 뒤에 있다.

단일준위일정작성프로그람은 보다 높은 준위에 있는 거래에 대해서는 하나도 모른다. 따라서 거래는 다음의 위치에 놓이게 된다.

- 엄격히 보다 낮은 준위로부터 제일 먼저 요구한 거래앞에
- 보다 높은 준위(자기가 있는 준위를 포함하여)로부터 오는 거래호출뒤에

이러한 일정작성알고리듬의 요구는 다음의 3가지 부분으로 구성되여 있는 순서도장(orderstamp)에 부합된다.

- 체계낮음박자로부터 유도되는 시간도장
- 거래의 접근클라스
- 순서번호

두 순서도장들이 시간도장과 같다면 접근준위가 보다 높은 준위 순서도장이 먼저 발생되였다고 본다. 만일 두 순서도장들이 시간도장과 접근준위가 같다면 순서번호가 보다 낮은 준위 순서도장이 먼저 발생하였다고 본다. 순서번호는 하나이상의 높은 준위 거래에 지연이 같은 시작시간을 할당하는 경우에 주소로 된다. 그러므로 순서도장은 거래에 대하여 완전한 순서를 만든다.

순서도장을 계산하기 위하여 매개 일정작성프로그람은

- 자기 준위에서 능동인 거래에 대한 최초의 시간도장(ETS)
- 최초의 보다 낮은 준위의 시간도장(ELTS)

를 보관한다.

새로운 거래의 시작시간은 다음과 같이 계산한다. 거래의 접근클라스에서 일정작성프로그람은 처리중에 있는 자기의 ELTS 를 갱신하고 있는 동안에 가장 낮은 준위 접근클라스로부터 시작하여 (격자형의)접근클라스를 우로 올라 가면서 보다 낮은 모든 준위의 ETS와 ELTS를 조사한다. ELTS를 갱신하자면 거래는 최소단위로 되여야 한다.

이 ELTS가 존재한다면 새로운 거래에 대한 시간도장으로 리용될것이며 그렇지 않으면 현재의 박자값을 리용할수 있다. 이때 일정작성프로그람은 접근클라스와 순서번호를 첨부하여 거래의 시작시간을 나타내는 순서도장을 창조한다.

자료항목의 새로운 판본에 대한 쓰기시간도장은 이 자료항목을 쓰는 거래의 시작시간으로 된다. 어떤 거래의 읽기시간은 그의 시작시간으로 된다. 따라서 순서도장의 측면에서 볼 때 거래의 모든 조작은 동시에 발생할수 있다.
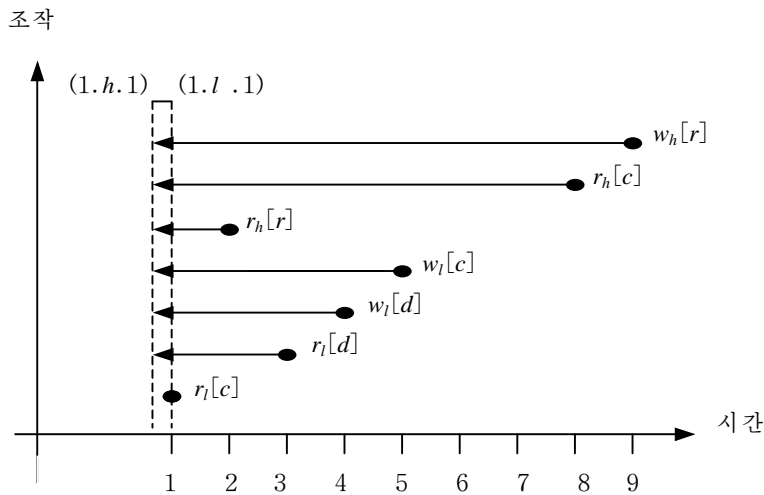
그림 16-2. MVTO-SS에서 시간도장의 할당

제16장 3절 2의 실례에서 거래 T_l에는 시작시간(1:l :1)이 할당되며 박자 2에서 그의 첫번째 조작이 발생되였다고 하여도 거래 T_h는 시작시간을 (1:h:1)로 잡는다. c.1과 d.1의 쓰기시간도장들이 모두 (0:l :1)이라고 하자. T_l이 이 두개의 자료항목들을 갱신할 때 c.2와 d.2인 새로운 판본들은 쓰기시간도장(1:l :1)을 수신한다. 따라서 T_h는 낡은 판본 c.1과 d.1만을 읽을수 있고 낡은 값 c:300, d:800에 근거하여 상환능력평가지표 A를 주게 된다. 그림 16-2는 거래에 있는 모든 조작들을 동일한 순서도장으로 넘기는 방법을 보여 주고 있다. 점선은 조작의 순서가 작성되여 실행되는 시간을 나타낸다. 화살표는 조작과 관계되는 시간도장을 가리킨다. 설명을 위하여 순서도장 (1.h.1)과 (1.l.1)에 각각 시간선도를 그었다.

4. MVTO-SS 의 정확성

병행조종알고리듬들은 언제나 엄밀한 분석을 하여야 하며 보안에 대한 사고가 부족한 경우에는 더욱 그러하다. 따라서 [78] 또는 [60]에서 론증된 MVTO-SS의 직렬화가능성정리증명을 요약하여 설명하려고 한다.

정리: MVTO-SS 에 의하여 생성된 처리순서들은 1회복사직렬가능하다. 이 처리순서들은 순서도장의 순서로 직렬로 실행하는것과 동등하다.

MVTO-SS 처리순서의 모든 읽기조작은 순서도장에 따라 직렬로 실행할 때의 자료항목과 같은 판본을 읽는다는것을 보여 주어야 한다.

이것을 여러판본자료기지에서 론의하고 단일판본자료기지우에서 직렬로 실행하는것과 등가적이다는것을 증명하자. 여러판본자료기지의 거래를 직렬로 실행할 때 거래는 언제나 읽기시점보다 앞에 있는 자료항목의 가장 최근의 판본을 읽는다. 따라서 단일판본과 여러판본자료기지는 직렬일정작성에서 차이가 없다.

시간도장에 의하여 정의되는 직렬일정작성에서 거래 T_1가 거래 T_2에 의하여 써진 자료항목 x 의 판본을 읽는다고 하자. 이때 MVTO-SS일정작성에서 T_1가 T_2에 의하여 써진 판본도 읽는다는것을 보여 주어야 한다. 반대로 T_1 가 또 다른 거래 T_3 에 의하여 써진 x 의 판본을 읽는다고 가정하자.

T_1는 T_2보다 앞에서 작성된 거래로부터만 자료항목을 읽을수 있다. 따라서 다음의 3가지 경우를 조사하여야 한다.

1. $T_2 < T_3 < T_1$: T_2보다는 뒤에서, T_1보다는 앞에서 T_3을 선택한다. 그러나 이때 T_1는 직렬일정작성에서와 마찬가지로 T_3에 의하여 씌여 진 x의 판본을 읽을것이다. 이것은 T_1가 직렬일정작성에서 T_2로부터 오는 x를 읽는다고 한 가정에 모순된다.
2. $T_3 < T_2 < T_1$: T_3은 T_2보다는 앞에 그리고 T_2은 T_1보다는 앞에 선택한다. T_1가 T_3으로부터 또는 x를 읽는다면 T_1의 읽기조작이 실행될 때 T_2은 x를 쓰지 못하였다. T_2이 쓰기할수 있는 자료항목이면 T_1가 다 읽을수 있으므로 T_1는 T_2 보다 접근클라스가 높아야 한다. T_1와 T_2이 동시에 능동이고 T_1가 T_2보다 후에 선택되였기때문에 T_1는 엄밀하게 보다 높은 준위 클라스안에 있을수 없다. 따라서 T_1와 T_2, x는 동일한 클라스안에 있어야 한다. T_1가 T_3에 의하여 씌여 진 x의 판본을 읽을 때 이 판본에 따라 읽기시간도장은 T_1의 순서도장으로 갱신된다. T_2이 이후에 x의 새로운 판본을 쓰려고 시도할 때에는 새로운 판본의 쓰기시간도장은 T_2의 순서도장으로 될것이며 T_3 에 의하여 써진 x에 대한 판본의 쓰기시간도장과 읽기시간도장사이에 놓여 있게 된다. 이것은 충돌로 되므로 MVTO-SS는 T_2을 포기할것이다. 따라서 T_2은 직렬처리로 될수 없고 T_1는 T_2이 쓴 x의 판본을 읽지 말았어야 할것이였다.
3. $T_3 < T_1, T_2$은 선택되지 않았다. T_1는 교차식처리순서로 T_3에서 x를 읽는다. 그것은 T_1가 x를 읽을 때 T_2의 순서가 정해 지지 않았기때문이다. T_1가 직렬순서로 작성된 T_2의 x판본을 읽을 때에 T_2은 T_1의 순서도장보다 앞에 있는 어떤 순서도장을 받아야 한다. MVTO-SS 는 T_2이 엄격히 T_1보다 높은 준위 클라스로 되여 있을 때에만 이것을 하게 된다.

한편 T_1가 T_2에 의하여 써진 자료항목을 읽도록 하기 위해서는 T_1 의 접근클라스는 T_2에 비하여 높아야 한다.

여기에서도 T_1가 교차식순서로 T_3에서 x를 읽는다는 가정은 모순으로 되였다.

제4절. 직렬화 불가능한 병행조종

직렬화가능성은 여러준위보안과 결합하여 별로 요구하지도 않는 병행조종알고리듬을 억지로 작성하게 하였다.

- 최량적인 MLS 일정작성에서는 언제나 높은 준위의 거래만이 서로 다른 접근클라스들과의 충돌로 실패하게 된다.
- MVTO-SS 순서작성에서는 서로 다른 접근클라스와 교차하는 충돌이 있다면 높은 준위거래는 언제나 자료항목의 보다 낡은 판본을 읽어야 한다.

여러준위보안, 일관성(직결가능성), 리용가능성사이에는 물론 서로 끌어 당기는 관계가 존재한다(그림 16-3).

지금까지는 보안과 일관성만 중심에 놓고 고찰하였고 높은 준위의 사용자들이 리용가능성은 거의 론의하지 않았다.

이제부터 여러준위보안자료기지에서 높은 준위사용자들을 위한 직렬화 불가능한 병행조종전략들을 검토한다.
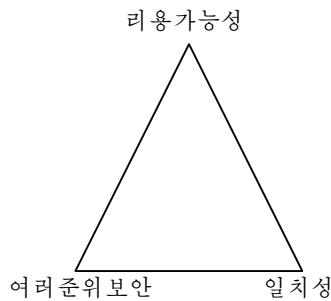
그림 16-3. 자료기지특성

다음과 같이 여러 판본정확성속성을 보다 약화시키면 대역적인 직렬화가능성으로 될수 있다.

- **단일준위직렬화가능성:** 단일접근클라스안에서 거래를 교차식으로 실행하자면 반드시 직렬가능하여야 한다. 실제적으로 단일접근클라스안에서 여러준위보안을 끌어 들이지 않고도 완정성제한을 임의로 적용할수 있다.
- **단일준위읽기일관성:** 거래가 완정성제한에 의하여 관계되는 보다 낮은 준위 접근클라스로부터 자료항목을 읽을 때에는 반드시 일치한 값들의 모임으로부터 자료항목을 읽어야 한다.
- **진전:** 자료항목의 개별적판본이 일단 거래에 의하여 읽혀 졌다면 바로 그 첫번째 거래에 의존하고 있는 어떤 거래도 자료항목의 이전 판본을 읽어서는 안된다.

[9]에서 언급된 다음의 일정작성알고리듬은 2-상잠금과 갱신된 MVTO 알고리듬을 결합하고 있다. 이 알고리듬은 높은 준위사용자들에게 보다 개선된 봉사를 제공하는 순서알고리듬의 실례로는 되지만 대역적인 직렬화가능성은 고려하지 않고 있다. 다시한번 여러판본자료기지를 론의하자.

- 매개 거래에는 시작시간도장이 하나뿐이며 완료된다면 위임시간도장(commit timestamp)은 하나이다.
- 매개 자료항목에는 여러개의 판본이 있다.
- 자료항목의 매개 판본에는 쓰기시간도장이 있다.
- 매개 쓰기조작은 자료항목에 대한 새로운 판본을 창조한다.
- 거래가 성공적으로 끝나면 새로운 판본이 자료기지에 영구적으로 들어 간다. 새로운 판본의 쓰기시간도장은 거래의 완료시간도장으로 된다.

MVTO-SS에 대하여 두가지 중요한 차이점이 있다. MVTO-SS 에서 거래는 순서도장에 관하여 하나의 실례로써 발생한다. 그러나 이 방법에서 거래는 자기의 시작시간으로부터 완료시간까지의 지속시간을 가진다.

다음으로 MVTO-SS 에서는 쓰기시간도장이 거래에 대하여 앞에 있는 시작시간이였다. 그러나 이 방법에서 쓰기시간도장은 새로운 판본이 자료기지에 들어 온 시간을 반영하고 있다.

거래를 질문(읽기만 한다.)과 갱신(쓰기도 포함될수 있다.)으로 구분할수 있다. 거래의 읽기시점은 자기의 시작시간으로 된다. MVTO을 리용하여 질문을 한다. 매개 입력의 요구는 쓰기시간도장이 질문읽기시점보다 앞선 자료항목의 가장 최근의 판본으로 넘어 간다.

갱신들은 아래읽기를 할수 있지만 자기의 준위에서만 쓰기할수 있다.

2-상잠금은 갱신준위에서 읽기 및 쓰기조작들을 순서화한다. 엄격히 보다 낮은 준위의 자료항목을 읽을 때 질문에도 MVTO 알고리듬을 쓸수 있다.

이 일정작성알고리듬에서 자료항목의 새로운 판본에 대한 쓰기시간도장은 갱신이 완료되기전에는 할당되지 않기때문에 이후의 쓰기에 의하여 읽기는 반드시 유효로 된다. 따라서 새로운 판본의 쓰기시간도장은 새로운 판본이 자료기지에 입력될 때에 능동인 임의의 거래의 읽기시점에 비하여 뒤떨어 지게 될것이다.

또한 거래에서 모든 읽기조작들은 동일한 시간도장(시작시간)으로 넘어 가며 모든 쓰기조작들은 동일한 시간도장(완료시간)으로 넘어 간다. 한 클라스범위에서 2-상잠금은 조작들이 직렬가능한 순서를 담보한다.

2-상잠금에서 높은 준위 거래 T_h가 낮은 준위 자료항목 x를 읽고 낮은 준위 거래 T_l가 x를 쓸 때 발생하는 충돌만은 해결할수 없다.

- $r_h[x]$가 $w_l[x]$보다 앞에서 발생하면 낮은 준위 쓰기조작이 x의 새로운 판본을 창조하므로 여러판본자료기지에서 충돌이 없게 된다.
- $w_l[x]$가 $r_h[x]$보다 앞에서 발생하고 T_h가 T_l에 의하여 씌여진 x의 판본을 읽는다면 어떤 등가적인 직렬순서로 T_h는 T_l보다 뒤에 그리고 x의 새로운 판본을 쓰는 이후의 그 어떤 낮은 준위 거래보다는 앞에 나타나게 된다.

바로 그렇기때문에 순서작성자가 다음과 같은 순서를 허용할 때만 여러판본자료기지에서 거래호출에 대한 교차식실행이 직렬가능할수 있다.

$$w_1[x.i]\ \ r_o[x.i]\ \ w_2[x.i]\ \ w_2[y.k]\ \ r_0[y.k]$$

여기서 T_0은 높은 준위 거래이고 T_1과 T_2 은 낮은 준위 거래이다.

어떤 등가적인 직렬순서에서나 T_0은 T_2보다 앞에 있어야 하므로 T_1로부터 x를 읽을수 있고 T_2보다 뒤에 있으므로 T_2로부터 y를 읽을수 있다.이것은 명백히 모순으로 된다.

T_0의 모든 읽기조작들은 동일한 읽기시점을 리용한다. 이 읽기시점이 T_2의 완료시간보다 앞에 있다면 T_0은 판본 $y.k$를 읽을수 없다.이 읽기시점이 T_2의 완료시간보다 뒤에 있다면 T_0은 $x.i$ 또는 X에 대한 이후의 판본을 읽는다.

이 절에서 제시한 일정작성알고리듬은 단일준위직렬가능한 실행을 생성한다. 그러나 갱신된 MVTO는 1회복사직렬가능이 아닌 순서들을 허락한다.

하나의 낮은 준위 자료항목 x,두개의 높은 준위 자료항목 y와 z의 거래를 고찰하자.

T_3:	$w_3[x]c_3$	low
T_4:	$r_4[y]r_4[x]w_4[y]c_4$	high
T_5:	$r_5[z]r_5[x]w_5[y]c_5$	high

그리고 이때 실행순서는 다음과 같다.

$$r5[z]w3[x]c3r4[y]r4[x]w4[y]c4r5[x]w5[y]c5.$$

그림 16-4에서는 갱신된 MVTO에서 이 3개의 거래에서 조작의 읽기시점과 쓰기시간도장을 계산하는 방법을 보여 주고 있다.
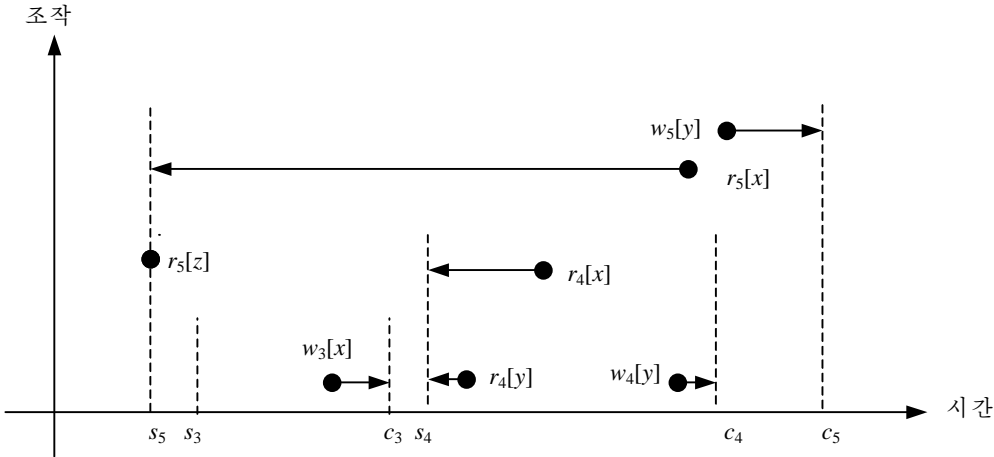
그림 16-4. 갱신된 MVTO에 의한 일정작성조작

단일판본자료기지에서 어떤 등가적인 직렬실행은 다음과 같다.

- T_5는 T_3보다 앞서야 한다. 그렇지 않으면 T_5은 T_3에 의하여 씌여 진 x의 판본을 읽게 되지만 MVTO순서에서 T_5은 X의 낡은 판본을 읽는다.
- T_3은 T_4보다 앞서야 한다. 그렇지 않는 경우에는 T_4는 T_5에 의하여 씌여진 y의 판본을 읽을수 없다. MVTO순서에서 T_4는 T_3으로부터 x를 읽는다.
- T_4는 T_5보다 앞서야 한다. 그렇지 않으면 T_4는 T_5에 의하여 씌여진 y의 판본을 읽게 될것이다. MVTO 순서에서 T_4는 y의 낡은 판본을 읽는다.

바로 그렇기때문에 그림 16-4의 실행과 같은 효과를 가지는 단일판본자료기지에서 T_3, T_4, T_5의 직렬실행은 없을수 있다.

단일판본자료기지에서 MVTO는 그림 16-5에서 보여 준것처럼 진전성을 지원하지 않는다.

하나의 낮은 준위 자료항목 *X*, 두개의 높은 준위 자료항목 *y*와 *Z*의 거래가 있다.

T6:	w6[x] c6	low
T7:	r7[x] w7[y] c7	high
T_8:	$r_8[z]$ $r_8[x]$ $w_8[y]$ c_8	high

이때 실행순서는 다음과 같다.

$r_8[z]$ $w_6[x]$ c_6 $r_7[x]$ $w_7[y]$ c_7 $r_8[x]$ $w_8[y]$ c_8.

그림 16-5는 이 3개의 거래에서 조작의 읽기시점과 쓰기시간도장을 계산하는 방법을 보여 주고 있다. 거래 T_7과 T_8은 x의 값에 의존하는 y의 판본을 쓴다. T_8은 x의 첫번째 판본을 읽고 T_7은 z의 두번째 판본을 읽지만 T_8보다 앞에서 y를 쓴다. 그렇기때문에 진전성을 위반할 때 y의 보다 최근의 판본은 T_7에 의하여 써진 판본에 비하여 낡은 X의 값에 의존한다.
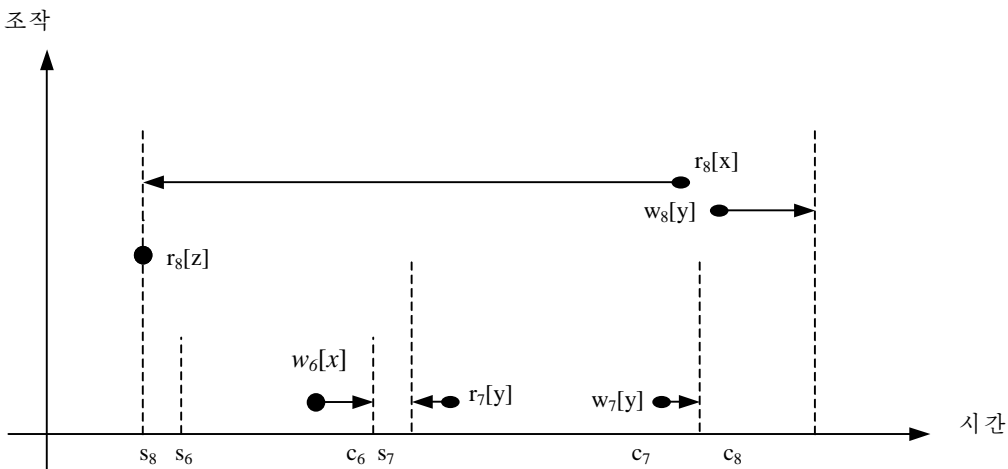
그림 16-5. 진전성을 위반하는 실행

결론

MLS자료기지에서의 병행조종은 어딘가 좀 억지로 선택하게 하는감을 준다.

- 직렬화가능성과 같은 엄격한 MLS보안방책들과 엄격한 병행조종기준을 시행할수 있다. 이때 최량적인 병행조종 그리고 MVTO-SS를 적용가능한 일정작성알고리듬은 높은 준위의 주동체들을 명백히 불리한 위치에 놓이게 한다.
- 보안방책들을 완화시킬수 있다. 실례로 대역적인 일정작성프로그람을 허용하면 표준일정작성알고리듬들중의 하나를 리용할수 있다. 이때 일정작성자는 신용 받는 주동체로 될것이며 잠복통로를 감시하는 기타 기구들이 리용되게 된다.
- 병행조종기준척도를 완화시키면 높은 준위의 사용자들에게 보다 개선된 봉사를 제공할수 있다. 이것은 가장 현실적인 결정이라고 볼수 있다. 실례로 이 절에서 서술된 일정작성알고리듬은 Trusted Oracle에 의하여 개발되였다.

이 장의 문헌안내

병행조종에 대한 교과서로는 [16]이 표준적이라고 볼수 있다.

MLS 자료기지에서 병행조종에 대한 최근의 조사에 의하면 [10]은 이 분야에 대한 충분한 견해를 준다고 말할수 있다.

또한 [60]에는 이 분야에 대한 가치 있는 SIR 보고가 있다.

[9,68,78,144]와 같은 연구론문들을 직접 보면 문헌조사를 심화시킬수 있다.

련습문제

1. T_1가 높은 준위거래이고 T_2이 낮은 준위거래라고 하자. 일정작성프로그람은 최량적인 병행조종을 리용한다. T_1가 처리를 완료하자면 다음의 경로중에서 어느것을 선택하여야 하는가.

   $r_1[x]r_2[x]w_2[x]$start val_1 start val_2 end val_1 end val_2
   $r_1[x]r_2[x]w_2[x]$start val_1 start val_2 end val_2 end val_1
   $r_1[x]r_2[x]w_2[x]$start val_2 start val_1 end val_2 end val_1
   $r_1[x]r_2[x]w_2[x]$start val_2 start val_2 end val_1 end val_1

2. MVTO-SS 일정작성알고리듬에서 ELTS의 계산이 최소단위조작으로 되는 리유를 설명하시오. 일관성위반이 일어 날수 있다는것을 보여 주는 기타 실례를 구성하시오.

3. MVTO-SS 를 리용하여 그림 16-4와 그림 16-5의 실례에서 보여 준 조작들의 순서를 작성하시오. 작성된 조작들이 1회복사직렬가능하게 되는 리유를 설명하시오.

4. T_1, T_2, T_3이 각각 준위 U, C, S (U<C<S)의 거래라고 하자. 이 3개의 거래로부터의 조작들은 다음의 순서로 일정작성프로그람에 도착한다고 하자.

   $r_1[z]w_1[x]r_3[x]r_2[y]r_3[y]w_3[x]c_1r_2[z]c_3c_2$

   자료항목 x, y, z의 준위를 어떻게 알수 있는가?
   MVTO-SS를 일정작성알고리듬으로 리용하는 경우에 이 조작들이 실행되였다고 보아 지는 순서는 어떻게 되겠는가?

5. 여러준위보안자료기지에서 단일준위직렬화가능성과 단일준위읽기일관성을 실현하는 병행조종알고리듬을 정의하시오. 정의한 알고리듬은 다음의 정황에서 어떻게 처리하겠는가? 3개의 거래 T_1, T_2, T_3으로부터 오는 조작들이 다음의 순서로 일정작성프로그람에 도착한다.

   $r_3[x]r_1[x]w_1[x]c_1r_2[x]w_2[y]w_3[y]c_2c_1$

   T_1, T_2, T_3이 각각 U, C, S (U<C<S)준위에서 분류되여 있는 경우를 고찰하시오. 또한 T_1는 준위 U에 있고 T_2, T_3은 준위 C에 있는 경우도 고찰하시오.

6. 구축블로크로서 제16장 4절에서 고찰한 일정작성프로그람을 리용하며 작성하여야 할 거래순서가 미리 알려 져 있는 경우 1회복사직렬화가능성을 실현하는 일정작성알고리듬을 설계하시오[9].

제17장. 객체지향보안

이 마지막장에서는 콤퓨터보안에서의 새로운 연구방향의 전망과 그의 실현에서 제기될수 있는 문제점들을 검토한다. 객체지향체계는 통제된 호출 즉 방법호출을 위한 일반적인 기구로 나온것이며 한편 정보의 은폐는 층아래(the layer below)에로의 접근을 금지하기 위하여 나온것다. 따라서 객체지향체계에서 보안은 연구하여야 할 필수적인 과정으로 된다. 이 과정에는 객체모형을 구축한 보안에 관계되는 특징들의 일반적평가로부터 시작하여 객체지향보안을 실현하는 여러가지 방법들을 비교하기 위한 위임접근조종까지 포함되여 있다.

목적

- 객체지향의 기본모형을 간단히 소개한다.
- 객체지향이 보안체계의 설계를 어느 범위까지 고유하게 지원하고 있는가를 론의한다.
- 객체모형에서 여러준위보안을 시행하는 두가지 다른 방법들을 제시한다.
- 콤퓨터보안의 새로운 방향을 지적한다.

제1절. 리론적기초

접근조종을 관리하자면 보안방책에 개별적인 사용자들이 개인자료항목들에 접근할수 있는 방법을 반영하여서는 안된다고 이미 이야기한바 있다.

얼마간은 조종의 중간층들이 있어야 한다.

클라크-윌슨보안모형에서 처음으로 이 방법이 구체적으로 론의되였으며 보임새에 기초한 자료기지보안은 가장 새로운 실례였다.

본질적으로 다음과 같은 보안방책을 실현하려고 한다.

- 특정의 조작만이 자료항목에 접근할수 있다. 그리고
- 사용자들에게는 특정의 조작들만 실행할수 있도록 허용된다.

객체지향은 우리가 요구하고 있는 모든 특징들을 다 가지고 있는것 같이 보인다.

자료항목들은 객체들안에 존재한다.

객체대상안에 있는 자료항목은 이 대상을 정의한 방법(method)들을 통해서만 접근할수 있다. 이것이 과연 보안을 실현하는 적합한 토대로 되는가? 보다 정확히 표현한다면 보안공격을 막는데 이러한 개념들로 충분히 설명할수 있겠는가? 만일 충분히 설명할수 있다면 이러한 설명서에 따라 실제적으로 실현된것이 있는가?

제2절. 객체모형

객체지향체계설계에 대한 본질적인 개념들만 언급하기로 한다.

먼저 학술용어에 대한 두가지 주의를 이야기하자. 이 장에서 소개되고 있는 객체(object)들과 벨-라파둘라에서 본 접근조종모형의 객체(object)들을 혼돈하지 말아야한다. 접근조종의 테두리에서 《새로운》 객체들은 주동체와 객체가 다될수 있다. 다음으로 보안의 실천가라고 볼수 있는 독자들은 객체지향공동체가 보편적으로 합의된 학술용어를 따라 세우지 못했다는 사실로부터 어느 정도 위안감을 가질수 있다. 따라서 아래에서 쓰이는 표현은 객체지향설계에 관한 모든 원천들에서 일치하지 않을수도 있다. 그러나 객체지향방법들에서 중요한 개념들에는 다음과 같은것들이 있다.

- 객체: 객체들은 속성(구체례변수(instance variable))들과 방법들로 이루어 져 있다.객체지향모형화는 독자들이 자기의 자료를 구성하는 방법과 조작하는 방법을 동시에 생각할것을 요구한다. 이것은 구도의 보안에 착수하는데서 그릇된 출발점이라고 볼수 없다. 자료기지구도의 설계가 틀리면 보안을 필요없이 복잡하게 만드는 하나의 원인으로 된다는것을 앞장들에서 언급하였다.
- 값: 속성들이 취할수 있는 값들은 다시 객체들로 된다.
- 형: 모든 객체는 자기가 소속되는 형(클라스)을 가진다. 객체는 자기 클라스의 구체례(instance)로 된다고 말할수 있다. 형은 다시 객체로 된다. 하나의 클라스에서 모든 객체들은 그우에서 동일한 조작들을 실행할수 있다는 의미에서 기능적으로 등가이다.
- 클라스계층과 계승(class hierachy and inheritance): 형들은 계층을 형성한다. 객체는 자기의 클라스와 뿌리클라스객체까지의 매개 상위클라스(superclass)로부터 속성과 방법들을 계승한다. 부분클라스는 자기의 초클라스로부터 계승된것들외에도 속성과 방법들을 가질수 있다.
- 원시객체(primitive object): 원시객체들은 수값, 문자, 식별자로 될수 있으며 식별자들은 부분클라스를 가지지 못할수도 있다.
- 방법(method): 어떤 객체에 대한 방법은 그 객체에 실행될수 있는 조작들만이 된다. 방법은 자기의 객체에서만 값들을 변화시킬수 있다.객체는 자기의 방법들에 의해서만 관측되고 변경될수 있으며 외부접근은 할수 없다. 방법들은 통보문(방법요구)을 보내여 다른 객체(그리고 그의 방법)들에 접근할수 있다. 매개 클라스는 자기의 형에 대한 새로운 실례를 창조하는 create 방법을 가지고 있다.
- 통보문: 통보문들은 객체들이 통신할수 있는 유일한 수단이다. 통보문에는 방법호출의 인수(파라메터)들이 포함되여 있다.
- 정보은페: 매개 객체는 자기의 국부주소공간을 가지고 있다.

콤퓨터보안에서 기본은 서로 다른 실체에 속하는 기억구역을 차페시키는것이다. 객체지향체계들은 개별적인 객체들의 작은 덩어리(granularity)로 차페를 실현한다. 이러한 각도에서 보면 정보은페는 강한 보안의 기초로 된다. 그림 14-3의 실례에서 관계 Diary를 객체지향으로 실현하면 속성이 Name,Dest,Flight,Status인 클라스객체 Journey와 Journey-read, Journey-update, Journey-create의 방법으로 구성되게 된다.

관계의 무이들은 그림 17-1에서 보여 준것처럼 이 클라스객체의 구체례에 대응한다.

이 구체례객체들은 자기의 클라스객체로부터 계승된다. 즉 방법 Journey-read 를 객체(Alice, Mon, GR123, private) 에 적용할수 있다. 이 객체에서 Journey-read외에 객체에 대하여 다른 읽기조작은 없다.

려행사가 자료기지체계에 가입되면 체계가입등록프로그람은 다른 모든 객체와 작용하게 될 TravelAgent 라는 객체를 창조한다. 객체지향관계 Diary 에 새로운 기입항목을 넣기 위하여 객체 TravelAgent는 클라스객체 Journey에 통보문을 보내여 새로운 구체례객체가 창조될것을 요구한다. 클라스객체는 자기의 create방법을 리용하여 우의 동작을 진행하며 새로운 객체에 대한 식별자를 포함하고 있는 되돌림값도 보낼수 있다(그림 17-2).
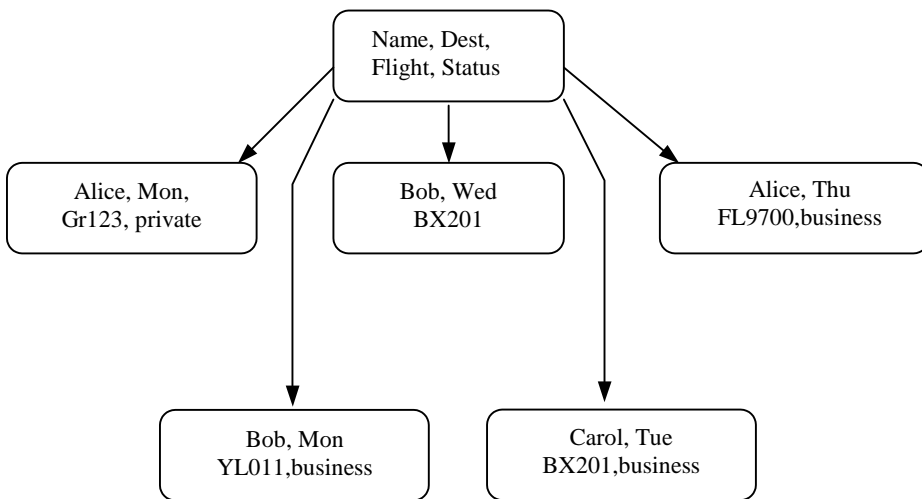
그림 17-1. 그림 14-3의 관계 Diary의 객체지향실현
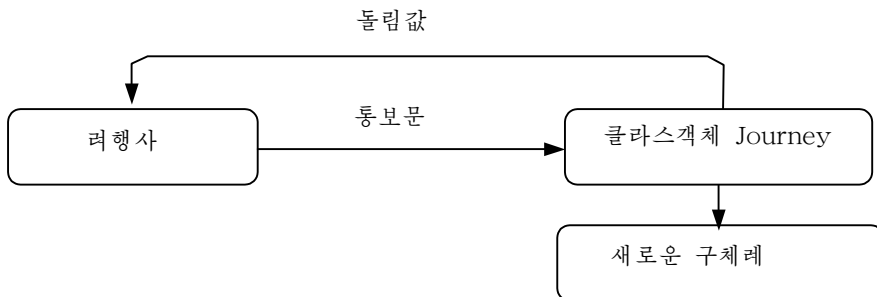
그림 17-2. 클라스 Journey에 대한 새로운 구체례의 창조

그러면 TravelAgent는 이 객체에서 Journey-update 방법을 리용하여 해당한 구체적인 려행봉사에 들어 갈수 있다.

제3절. 객체모형에서 보안

객체지향체계들은 보안을 지원하는 일련의 쓸모 있는 특징들을 제공한다.

객체에서 자료를 교갑화함으로써 자료의 접근을 조종할수 있는 기회를 가진다. 자료항목(속성)들은 특정된 방법으로만 접근될수 있다. 정보은페는 외부적인 간섭으로부터 객체의 완정성을 보호한다. 우리의 콤퓨터모형에서는 객체지향보안이 여러 층에서 실현되였다.

- Trusted Mach는 보안핵심을 가지는 객체지향조작체계이다.
- Java와 그의 보안구성방식은 객체지향원리에 기초하여 구성되였다(제11장 6절).
- CORBA는 분산객체체계를 위한 보안구성방식이다(제10장 4절).

객체지향체계들은 계승의 측면에서 안전한가? 교갑화와 정보은페는 객체지향모형에서의 기본개념들이다. 누구나 다 자기의 객체들과 객체안에서의 방법을 정의할수 있으며 자기의 보안방책을 획득하기 위하여 일부 방책객체를 첨부해 줄수도 있다.

일감이 자기의 목적에 맞게 실행되였다면 정보은페는 안정성에 주의를 돌려 자기체계의 완정성을 보호할것이다. 정보은페야말로 두 세계에서 가장 좋은것을 다 받아 들였다고 볼수 있다. 객체에 의해 보안방책을 기술하는데 특징이 풍부한 환경이 조성되지만 정보은페에 의해 이러한 모든 보안방책들을 시행하는데 도움을 주는 포괄적인 보안기구가 실현된다고 볼수 있다.

방어자들과 공격자들이 객체지향체계가 제공하는 추상층에서만 체계를 고찰하는 조건에서는 이러한 견해는 정확하다고 말할수 있다. 우에서 이야기한바와 같이 정보은페는 이 층이 놓여 있는 실제적인 체계에 의존한다. 층아래를 호출하는 공격자는 일반적으로 보안을 약화시킬수 있다.

Java 보안의 력사(제11장 6절과 [95])를 돌이켜 보면 이러한 주장을 안받침하고 있는 실례들이 풍부하다. 일반적으로 대부분의 객체지향체계들은 안전하게 동작하도록 설계되지 않았다는것을 잊지 말아야 한다.

정보은페를 실현하여도 고의적인 공격을 막아 낼수 있는 보안기구로는 되지 못한다. 설계자들은 집요한 공격자가 조종을 우회하여 따라 올수 있는 모든 회로적인 경로들을 차단하는데는 관심을 돌리지 않았다.

쏘프트웨어작성자들이 코드의 효률을 높이기 위하여 리용하는 보다 낮은 층에로의 접근을 주는 엿보기구멍들이 알려 져 있을수도 있으며 이것을 [155]로부터 여기에 관계되는 대목을 다음과 같이 인용할수 있다.

> Ada, Modula-3 과 같이 형이 강한 언어라고 하여도 체계에 없는 형을 만드는 경우 프로그람작성자들은 흔히 형체계의 엿보기구멍을 리용할 필요가 있다는것을 알게 된다.

실천에서는 매 경우에 객체지향체계가 표준방안에 비하여 특징이 풍부하지만 담보가 적은 해결방도를 주고 있는가를 검사하여야 한다.

모든 봉사가 다 보호를 위하여 설계된 보안특징들을 발휘하고 있는것은 아니다.

제4절. 객체지향체계에서의 MAC

위임접근조종은 보안표식들을 리용하여 어떻게 주동체가 객체들에 접근하는가를 조절하고 있다.

객체모형에서 《주동체》와 《객체》는 어떤 기능을 수행하는가?

간단히 말하여 MAC-주동체는 방법의 요구를 전송하는 객체이고 MAC-객체는 방법이 실행되는 객체이다. 사용자가 체계에 가입되면 체계가입프로그람은 사용자를 나타내는 객체를 창조하고 사용자의 보안통과허가를 수신한다.

MAC 방책들을 객체모형으로 다시 표현하면 하나의 객체로부터 오는 정보는 동일한 또는 보다 높은 보안준위에 있는 객체에로만 흐를수 있다.

정보흐름의 운반수단은 통보문과 방법호출의 돌림값들이다. 정보의 흐름은 객체가 자기 상태를 변화시키거나 새로운 객체를 창조할 때 진행된다.

1. 객체의 표식화

객체모형에서 표식화될수 있는 객체의 양상은 풍부하다.

객체자체는 물론, 그의 속성, 속성의 값, 방법,통보문들이 양상으로 될수 있다.

객체의 창조자의 준위나 이 객체의 매개 속성을 찾아 나가면 표식화될수 있는 모든 양상을 알아 낼수 있을것이다. 이 준위들은 객체 또는 속성의 준위와 반드시 일치하는것은 아니다. 여러준위객체에는 서로 다른 보안준위의 양상들이 있다. 단일준위객체에는 객체와 그의 모든 양상에 적용되는 단일보안표식이 있다.

표식화는 클라스계층과 일치되여야 하며 또한 객체의 여러가지 양상들사이의 관계와도 일치하여야 한다. 객체들이 다음과 같은 방책들을 제기한다면 자기의 클라스로부터 오는 속성과 방법들을 계승한다.

- 속성의 보안준위는 모든 부분클라스에서 이 속성의 보안준위보다 우에 있다.
  만일 클라스객체를 조사하는 방법으로 속성의 존재를 알아 내려고 한다면 속성의 존재를 구체례객체에 은페시키려고 시도하는것은 무의미하게 된다. 이것은 실제적으로 잠복통로를 창조하게 될것이다.
- 방법의 보안준위는 모든 상위클라스에서 이 방법의 보안준위보다 우에 있다.
  방법이 클라스객체에서 《비밀》로 표식화되였어도 일반사용자(unclassified users)들이 어떤 구체례객체에서 그것을 실행시켜 그 기능을 알수 있다고 하면 무엇때문에 앞에서 그것을 비밀이라고 표시하였겠는가? 이와는 달리 방법이 모든 특정의 구체례가 아니라 일반적인 클라스객체에서 실행할수 있게 해주면 아주 합리적일것이다.

이 두가지 방책들은 보안을 서로 다른 측면에서 고찰하면 클라스계층안에 있는 보안표식들도 서로 다른 각도에서 분석할수 있다는것을 보여 준다. 더우기 다중계승을 지원하는 체계라면 문제가 제기된다. 이 경우에 하나의 객체는 하나이상의 클라스의 구체례로 될수 있다.

여러준위객체에서 속성들은 여러가지 보안표식들을 가질수 있다.

다중관계자갱신충돌(Multi-party update conflict)은 이러한 정황에서 제기되는 문제로 널리 알려 져 있다. 서로 다른 준위에 있는 두개의 주동체가 동일한 속성을 갱신한

다. 《낮은》주동체는 《높은》값이라는 존재를 반드시 알아야 하는가? 다중구체례제시에서는 잠복통로를 피하는 방법으로 이 문제를 해결하고 있다.

이 문제는 단일준위객체에서는 제기되지 않지만 서로 다른 보안준위에서 동일한 이름으로 객체들을 우연적으로 창조할수 있다는 문제는 여전히 남아 있다.

일치한 표식방책을 결정하자면 다음의 질문에 대답을 주어야 한다.

- 속성의 보안준위들은 객체의 보안준위보다 반드시 높아야 하는가?
- 객체의 보안준위들은 속성의 보안준위보다 반드시 높아야 하는가?

속성, 방법, 제한조건과 같은 객체의 양상을 표식화하는 리유는 좁은 의미에서는 객체 그자체의 준위에서 《현실적》인 보안요구(관계형자료기지와 비슷하다)를 표현하는데 있고 넓은 의미에서는 객체준위에서 잠복통로와 높은 준위정보의 간섭을 차단하는데 있다. 이러한 개별적인 잠복통로를 피하는 다른 방법이 널리 알려 져 있다.

만일 낮은 준위의 주동체가 높은 보안준위를 가지는 속성을 창조하였다면 그것은 자기의 존재를 알고 있으므로 낮은 준위주동체로부터 이 속성을 은페시킬 필요가 없다.

만일 객체지향모형화에 의해 응용자료들이 론리정연한 실체들로 구조화되여 간다는 주장에 동의한다면 여러준위객체들의 값을 질문하는것은 당연하다고 말할수 있다. 《론리정연한》실체는 반드시 단일한 보안표식을 가져야 한다.

2. 통보문 흐름조종

여기서도 객체지향모형에 따른다고 할 때 MAC는 객체들사이의 통보문흐름을 조종함으로써 실현되게 된다.

기입방책들은 객체표식들과 통보문을 넘겨 주겠는가를 결정할 때 방법의 형(읽기 또는 쓰기)에 의거한다. 방법의 호출은 통보문처럼 직접적으로 실현되며 자기의 기억기를 가지고 있지 않다. 방법의 호출은 객체를 리용하여 자료를 기억시켜야 한다.

강제접근조종은 객체들사이의 정보의 흐름을 규제한다.

객체 o_1로부터 o_2로 넘겨 준 통보문이 실제적으로 o_2의 상태 즉 o_2의 어떤 속성의 값을 변화시키는 경우에만 o_1로부터 o_2에로 정보가 흐른다.

이때 정보의 흐름은 다음과 같을수 있다.

- 정방향: 통보문의 파라메터를 통하여 방법호출을 내보내는 객체로부터
- 역방향: 방법호출에 응답하여 제공된 값을 통하여
- 이행: 정방향과 역방향흐름으로 된 임의의 사슬을 통하여
- 간접: 제3객체에서 내부상태의 변화를 통하여

간접적인 정보흐름을 막기 위하여서는 통보문호출이 호출된 객체만을 감시하는가 또는 그것이 바로 그 객체의 상태도 변화시킬수 있는가를 검사하여야 한다.

자조디아(Jajodia)와 코간(Kogan)에 의하여 제안된 통보문려파알고리듬[69]에서는 매개 객체에 함수 L에 의하여 정의된 하나의 보안표식을 가지고 있다.

일반적으로 보안표식들은 격자를 형성한다. $L(o_1)<>L(o_2)$ 로써 두 준위가 비교할수 없다는것을 나타내자.

고찰하는 방법들은 읽기(read), 쓰기(write), 창조(create)이다. 방법호출에서 매개 방법은 제한이 없는 상태(U) 또는 제한이 있는 상태(R) 중의 하나의 상태를 가진다. 간접적인 정보흐름을 조종하기 위하여 상태정보를 사용한다(사용자의 가입등록과 같은 체계를 초기적재하는데 요구되는 방법들은 제한이 없는 상태로 된다고 가정한다).

객체 o_1가 방법 t_1를 실행하여 방법 t_2이 실행되여야 한다고 요구한다면 t_1가 통보문 g를 객체 o_2에 보내는 정황을 통보문려과알고리듬으로 표현하자.

방법 t_2은 값 r를 o_1에로 되돌린다. 이 알고리듬은 정보가 보다 높은 준위의 객체에로만 흐르게 한다. 이 알고리듬은 통보문 g를 통과시키거나 차단시킬수 있으며 허용되지 않는 모든 방법의 호출에 대하여 령을 되돌린다.

통보문려과기는 우리가 볼수 있는 참조감시기에 불과하므로 방법이 객체안에서 실행되도록 허락해 주겠는가도 결정하여야 한다.

객체가 어떤 통보문을 그자체에 보내게 하여 이 통보문이 통보문려과기를 통과하는 경우에만 처리를 계속해 주게 함으로써 방법의 실행을 결정할수 있다. 이 경우 개별적인 객체들에 있는 방법들은 TCB의 부분으로 되지 않는다.

TCB를 보다 작게 한다면 담보가 보다 높은 준위 보안기구를 실현할수 있는 가능성이 열려 지게 된다.

통보문려과알고리듬
Case $o_1 \neq o_2$: /방법호출/
　　If $L(o_1) = L(o_2)$　　g를 통과시킨다; $s(t_2) := s(t_1)$;
　　If $L(o_1)$ <> $L(o_2)$　　g를 차단한다;
　　If $L(o_1) < L(o_2)$　　g를 통과시킨다; $r :=$ nil; $s(t_2) := s(t_1)$;
　　If $L(o_1) > L(o_2)$　　g를 통과시킨다; $s(t_2) := R$;
Case $o_1 = o_2$: /방법의 실행/
　　If t_1 = write
　　　If $s(t_1) = U$　g를 통과시킨다;
　　　If $s(t_1) = R$　g를 차단한다;
　　If t_1 = read　g를 통과시킨다;
　　If t_1 = create$(o_3, L(o_3))$
　　　If $s(t_1) = U$ and $L(o_3) \geqq L(o_1)$
　　　　　then g를 통과시킨다;
　　　　　else g를 차단한다;

그림 17-3은 통보문려과알고리듬이 간접적인 아래쓰기를 하지 못하게 하는가를 보여준다. 보안준위가 $L(o_3) = L(o_2) < L(o_1)$인 3개의 객체가 있다. 높은 준위 객체 o_1는 통보문 g_1를 낮은 준위 객체 o_2에 보내며 이때 g_1는 o_2의 상태를 변화시키지 않지만 o_2로부터 일부 파라메터들을 방법의 호출 t_2에 제공함으로써 낮은 준위 객체 o_3에로의 쓰기를 요구한다면 통보문 g_2을 o_3에 보내게 된다.

이 쓰기는 방법 t_3=write 로 요구한다면 o_3으로부터 그자체에로 통보문 g_3을 기동한다.

이 알고리듬의 실행단계는 다음과 같다.

단계1. 통보문 g_1: $o_1 \neq o_2$ 이고 $L(o_1) > L(o_2)$ 이므로 g_1 는 통과될것이며
　　$s(t_2) := R$
단계 2. 통보문 g_2: $o_2 \neq o_3$이고 $L(o_2) = L(o_3)$ 이므로 g_2 은 통과될것이며
　　$s(t_2) := R$
단계 3. 통보문 g_3: g_3을 o_3으로부터 그자체에 보내여 t_3 = write
　　$s(t_3) := R$ 이므로 g_3는 차단되며 쓰기는 실행되지 않을것이다.

이 알고리듬에서 이 구체례에 보내야 할 임의의 특수한 돌림통보문을 하나도 표현하지 않고 있다.
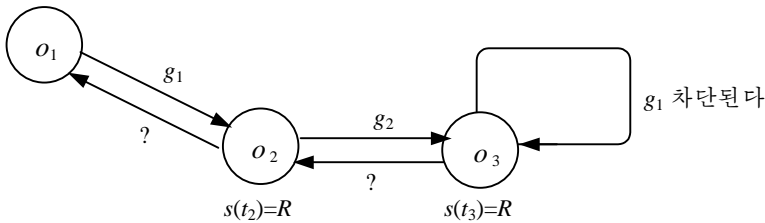
그림 17-3. 자조디아-코간모형에서의 중개접근

3. MLS 조작체계의 객체지향보안

밀렌-룬트(Millen-Lunt)모형[10]은 전통적인 MLS 조작체계우에서 객체지향 여러준위보안을 실현한다. 따라서 객체모형에 대한 개념은 벨-라파둘라모형에 대한 개념으로 넘어 간다. 방법호출은 처리 즉 벨-라파둘라의 용어로 표현하면 자기의 기억공간과 자기의 보안준위를 가지는 주동체로 된다. 방법접근에 의하여 접근되는 객체들은 벨-라파둘라의 의미에서 객체들이다.

- 객체가 방법의 요구를 받으면 통보문에서 방법호출을 처리하기 위한 목적에서만 가상주동체(virtual subject)가 창조된다.
- 가상주동체는 방법호출을 홈객체로 수신하는 객체를 가진다.
- 객체 o_1의 방법 t_1가 객체 o_2의 방법 t_2에 대한 요구를 보내면 두 방법 t_1와 t_2과 관련되여 있는 가상주동체 t_1와 t_2이 존재하게 된다.

표기는 앞절에서와 같다. 단일준위객체들과 격자형의 보안준위들이 있다. 함수 L은 매개 객체와 매개 가상주동체들에 대한 보안준위를 나타낸다.

가상주동체들과 그의 실제적인 객체(home object) 그리고 호출자들을 위하여 위임적인 보안방책을 형식화할수 있다. 통보문을 전송하는것을 통제할 방도가 없고 모든 호출이 실제적인 객체에서 결정되였기때문에 방법호출에 의한 돌림값을 처리하는 방책이 필요하게 된다. 이와 같은 방책은 객체가 어떤 환경에서는 령인 값을 되돌릴것을 요구한다.

MAC 방책들은 다음의 6가지 보안특성으로 이루어 져 있다.

규칙 1: 객체의 보안준위는 그의 클라스객체의 보안준위보다 높아야 한다.

규칙 2: 가상주동체의 보안준위는 주동체를 호출하는 준위보다 높다. 가상주동체의 보안준위는 또한 실제적인 객체준위보다 높다.

규칙 3: 가상주동체는 실제적인 객체에서만 방법들을 실행하거나 또는 변수들을 읽고 쓸수 있다.

규칙 4: 가상주동체는 보안준위가 실제적인 객체의 보안준위와 같은 경우에만 실제적인 객체에 쓰기할수 있다.

규칙 5: 가상주동체는 호출하고 있는 주동체와 보안준위가 같은 경우에만 돌림값을 그가 접근하고 있는 주동체에 보낼수 있다.

규칙 6: 최근에 창조된 객체의 보안준위는 창조를 요구한 가상주동체의 준위보다 높다.

그림 17-4에서 o_1은 방법의 호출을 o_2에 전송하는 객체라고 하자.

호출하고 있는 주동체 s_1의 보안준위는 실제적인 객체 o_1의 준위와 같다고 가정하자. 실제적인 객체 o_2을 가지는 새로운 가상주동체 s_2은 현재의 방법호출을 처리한다. 3가지 경우를 다 고려하면 규칙 2는 $\max(L(s_1), L(o_2))$의 준위, 즉 보다 높은 보안준위에서 가상주동체 s_2을 분류한다.

- $L(o_1) < L(o_2)$: $L(s_1) < L(s_2)$로 되기때문에 규칙 5는 요구한 접근의 형에 관계없이 o_2로부터 o_1의 되돌림값이 령으로 된다고 규정하고 있다.
- $L(o_1) \geqq L(o_2)$ 이고 읽기방법의 호출: $L(s_1) = L(s_2)$이므로 규칙 5는 적당한 값을 되돌리게 한다.
- $L(o_1) > L(o_2)$ 이고 쓰기방법의 호출: 규칙 4는 $L(s_2) > L(o_2)$이기때문에 쓰기방법을 차단한다.
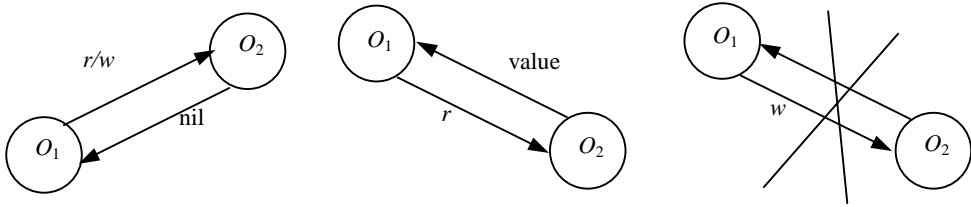
그림 17-4. 밀렌-룬트모형에서 중개접근
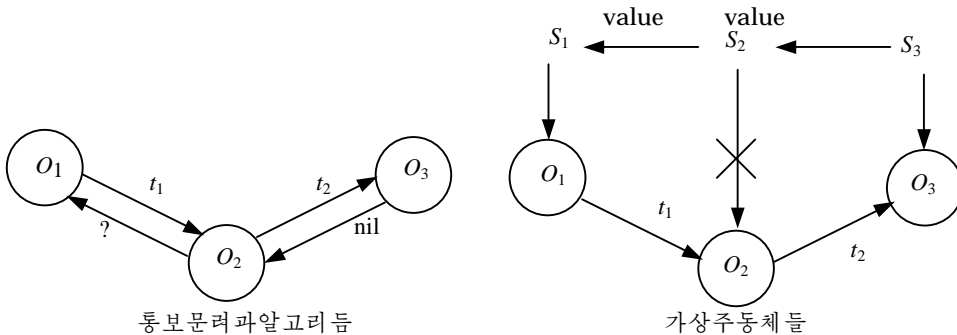
그림 17-5. 정보를 낮은 준위 중개자를 통하여 넘겨 주는 방법

밀렌-룬트모형으로 해석한다면 방법호출 t_1를 객체 o_2에 보내여 방법의 호출 t_2을 o_3으로 절환함으로써 객체 o_2을 변화시키지 않고 o_3은 어떤 값을 o_1에 되돌리는 방법으로 객체 o_1가 객체 o_3으로부터 오는 값을 읽을수 있다. 그것은 방법(가상주동체)들은 자기의 기억공간을 가지고 있기때문이다. 앞의 모형에서는 o_2은 맹목적인 중개자의 기능을 수행할수 없다.

따라서 호출조종이 서로 다르게 결정된 두 모형을 론의하여야 하는 정황이 존재한다. $L(o_1) = L(o_3) > L(o_2)$ 이면 두 모형을 그림 17-5처럼 결정할수 있다.

- 통보문려과알고리듬은 통보문나르개 t_1를 통과시킬것이다. 통보문나르개 t_2의 되돌림값은 자동적으로 령(그리고 $s(t_2) = R$) 으로 설정될것이다. 그러므로 첫번째 통보문에로의 되돌림값은 o_3으로부터 오는 정보를 포함하지 않는다.
- 밀렌-룬트모형에서 3개의 모든 가상주동체들은 같은 준위에 있다. 따라서 s_3은 어떤 값을 s_2에로 되돌리며 이것이 s_1에 넘겨 질수 있다. 가상주동체 s_2에는 자기의 실제적인 객체의 쓰기를 할수 없지만 그렇게 할 필요도 없다.

이 장의 문헌안내

객체지향보안자료기지의 현 실태에 대해서는 [25]에서 한개의 장으로 개괄되였다. 보다 깊은 내용을 보자면 연구론문들을 읽으면 될것이다.

처음에 [18] 과 [152]들을 읽으면 될것이다.

련습문제

1. 객체지향기본모형이 높은 준위 믿음성을 가지면서 접근조종기능이 풍부한 특징들을 제공할수 있겠는가? 이러한 요구가 정당하다고 납득시킬수 있는 조건들을 조사하시오. 조사해 보면 보안특징의 관리는 물론 공학적기구들이 포함되며 실현가능한 믿음성준위에 영향을 미친다는것을 알수 있다.

2 .형시행체계에서의 결함들이 정보은페에 기초하는 보안기구를 어떻게 약화시킬수 있는가를 보여 주는 실례연구를 진행하시오.(Java의 개발에서 얻은 경험은 이러한 하나의 자료로 된다.[95])

3. 속성들과 값들이 자기의 보안표식을 가지는 여러준위객체를 위한 일치한 표식방책을 론의하시오.
   - 속성에 대한 값의 보안준위가 속성의 보안준위보다 우에 있어야 하는가?
   - 속성의 보안준위는 속성에 대한 값의 보안준위보다 우에 있어야 하는가?

4. 여러준위객체를 단일준위객체로 표현할수 있다는것을 설명하시오.

5. o_1가 속성과 값들이 자기의 보안표식을 가지는 여러준위객체라고 하자. 객체 o_1는 값 o_2을 취하는 속성 a를 가진다.
   o_1, o_2, a 의 보안준위가 어떤 관계에 있을 때 새로운 객체를 창조하지 않고 o_1에 있는 속성 a의 값을 변화시킬수 있겠는가?

6. 객체 o_1가 창조요구를 o_2에 보냄으로써 형이 o_2인 객체 o_3을 창조하려고 하는 경우를 고찰하자. o_1, o_2, o_3 의 보안준위에 관계되는 통보문흐름모형에서 이 요구가 어떻게 처리되겠는가를 설명하시오.

7. 밀렌-룬트 방책에서 제3자의 객체 o_3에 쓰기를 요구함으로써 객체 o_1는 자기 준위에 있는 객체 o_2에로의 쓰기를 금지할수 있는가? $L(o_1) = L(o_3)$, $L(o_1) < L(o_3)$, $L(o_1) > L(o_3)$ 일 때 론의하시오.

참고문헌

1 E.Amoroso. *Fundamentals of Computer Security Technology*. Prentice-Hall International, Englewood Cliffs,nj,1994.
2 J.Anderson. Computer security technology planning study. Technical Report 73-51,U.S.Air Force Electronic Systems Technical Report,October 1972.
3 R.J.Anderson. and M.kuhn. Tamper resistance - a cautionary note. *In Second USENIX Workshop on Electronic Commerce,* pp 1-11,November 1996.
4 R.J.Anderson and F.A.P.Peticolas. On the limits of steganography. *IEEE Journal on Seleted Areasin Communications,*16(4) pages 474-481,February 1998
5 J.Arceneaux. Experience in the art of security. *Security Audit & Control - Review,* 14(4):12-16, October 1996.
6 European Computer Manufacturers Association. Commercially-oriented functionality class for security evaluation (COFC). Technical Report ECMA 205,December 1993.
7 Europan Computeer Manufacturers Association. Secure information processing versus the concept of product evaluation. Technical Report ECMA TR/64, December 1993.
8 D.Atkins, P.Buis, C.Nachenberg, A.B.Nelson, P.Phillips, T.Ritchey, T.Sheldon, and J.Snyder *Internet Security*. New Riders, Indianapolis,IN,2nd edition, 1997
9 V.Atluri, E.Bertino, and S.Jajodia. Achieving stricter correctness requirements in milti-level secure databases. In *Proceedings of the 1993 IEEE Symposium on Research in Security and Privacy,* pp 135-147,1993 .
10 V.Atluri, S.Jajodia, T.F.Keefe, C.McCollum, and R.Mukkamala. Multi-level secure transaction processing: Status and prospect. In P.Samariti and R.Sandhu, editors, *Database Security X: Status and Prospects*,1997.
11 D.B.Baker. Fortresses built upon sand. In *Proceedings of New Security Paradigms Workshop,* pp 148-153, September 1996.
12 D.Bell and L.LaPadula. MitreTechnical Report 2547 (Secure Computer System): Volume II. *Journal of Computer Security,* 4(2/3) pages 239-263,1996.
13 D.Bell and L.LaPadula. Secure computer system: Unified exposition and Multics interpretation. Technical Report ESD-TR-75-306, The MITRE Corporation, Bedford,MA,1975.
14 S.Bellovin. Security problems in TCP/IP protocol suite. *ACM Computer Communcations Review,* 19(2) pages 32-48, April 1989.
15 S.M.Bellovin and M.Merrit. Limitations of the Kerberos Authentication System. *ACM Computer Communcations Review,* 20(5) pages 119-132, 1990.
16 P.A.Bernstein, V.Hadzilacos,and N.Goodman. *Concurrency Control and Recovery in Database Systems.* Addison Wesley, Reading, MA,1987.
17 D.Berstis et al.IBM System/38 addressing and authorization. Technical Report GS80-0237,IBM System/38 Technical Development, 1978.
18 E.Bertino, L.V.Mancini,and S.Jajodia. Collecting garbage in multilevel secure object stores. In *Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy,* pp 106-120,1994.
19 K.J.Biba. Integrity consideration for secure computer systems. Technical Report ESDTR-76-372,MTR-3153, The MITRE Corporation, Bedford,MA,April,1977.
20 R.Bird, I.Gopal, A.Herzberg, P.Janson, S.Kutten, R.Molva,and M.Yung. The KryptoKnight family of light-weight protocols for authentication and key distribution. *IEEE/ACM Transactions on networking,* 3(1) pages 31-41,February 1995

21 B.Blakely. The emperor's old armour. In *Proceedings of the New Security Paradims Workshop,* pp 2-16, September 1996.
22 V.Bontchev. Possible macro virus attacks and how to prevent them. *Computer & Security*, 15(7) pages 595-626, 1996.
23 D.F.C.Brewer and M.J.Nash. The Chinese Wall security policy. In *Proceedings of the 1989 IEEE Symposium on Research in Security and Privacy,* pp 206-214,1989.
24 P.Brinch Hansen. *Operating Systems Principles* Prentice-Hall,Englewood Cliffs, NJ,1973.
25 S.Castano, M.Fugini, G.Martella, and P.Samarati. *Database Security*. Addison Wesley, Reading, MA,1994
26 CCEB. *Common Criteria for Information Technology Security Evaluation,*Version 2.0. May 1998.
27 CCITT. Recommendation X.509: The Directory - Authentication Framework, 1989. CCITT Blue Book, Volume VIII, Fascicle VIII.8.
28 D.B.Chapman and E.D.Zwicky. *Building Internet Firewalls*. O'Reilly & Associates, Sebastopol, CA,1995
29 W.R.Cheswick and S.M.Bellovin. *Internet Security*. Addson Wesley, Reading, MA,2nd edition, 1996
30 S.Chokhani. Trusted product evaluations. *Communications of the ACM*,35(7):64-76, July 1992.
31 Z.Ciechanowicz. Risk analyse: requirements,conflicts and problems. Computer & Security, 16(3) pages 223-232, 1997.
32 D.R.Clark and D.R.Wilson. A comparison of commercial and military computer security policies. In *Proceedings of the 1987 IEEE Symposium on Research in Security and Privacy,* pp 184-194,1987.
33 A.Clements. *Microprocessor System Design*. PWS-Kent Publishing Company, Boston, MA, 2nd edition,1992.
34 F.J.Corbato. On building systems that will fail. *Communications of the ACM* 34(9) pages 72-81 September 1991.
35 D.A.Curry. Improving the security of your Unix system. Technical Report ITSTD-721-FR-90-21, SRI Computer Science Laboratory,1990.
36 D.A.Curry. *Unix System Security.* Addison Wesley, reading, MA,1992.
37 C.J.Date. *An Introduction to Database System-Volume I.*Addison Wesley,Reading,MA,5th edition 1990.
38 M.De Blasi. *Computer Architecture*. Addison Wesley,Reading,MA,1990.
39 D.E.Denning, *Cryptograph and Security*. Addison-Wesley, Reading,MA,1982.
40 D. E. Denning, T. F. Lunt, R. R. Schell, W. R. Shockley, and M.Heckman. The SeaView Security Model. In *Proceedings of the 1988 IEEE Symposium on Research in Security and Privacy,* pp 218-233,1988.
41 W.Diffie and M.E.Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*,22:pages 644-654,1976.
42 W.Diffie, P.C.van Oorchot,and M.J.Wiener. Authentication and authenticated key exchanges. *Codes, Design and Cryptography*, 2: pages 107-125,1992.
43 S.Dreyfuss. Underground. Reed Books, 1997.
44 M.W.Eichin and J.A.Rochlis. With microscope and tweezers: an analysis of the Internet virus of November 1988. In *Proceedings of the 1988 IEEE Symposium on Research in Security and Privacy*, pp 326-343, 1988.
45 T.El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*,31(4) pages 469-472,1985.
46 C.M.Ellison,B.Frantz, B.Lampson, R.Rivest, B.M.Thomas,and T.Ylonen. *SPKI Certificate Theory,* March 1998. Internet Draft.
47 E.English. Network security under siege. *IEEE Computer*,29(3) pages 95-97, March 1996.

48 D.C.Feldmeier and P.R.Karn. UNIX password security- ten years later. In *Advances in Cryptology - CRYTO'89*, pp 44-63. Springer LNCS 435,1990.

49 D.Ferbrache. *A Pathology of Computer Viruses*. Springer-Verlag, London,1992

50 D. Ferbrche and G. Shearer. *UNIX Installation Security and Integrity*. Blackwell Scientific Publications, Oxford, 1992.

51 International Organisation for Standardization. *ISO 7498-2 Basic Reference Model for Open Systems Interconnection (OSI) Part 2: Security Architecture.* Geneva, Switzerland, 1988.

52 International Organisation for Standardization. *ISO/IEC 9075: Information Technology-Database Languages - SQL*. Geneva, Switzerland,1992.

53 W.Ford. *Computer Communications Security*.Prentice-Hall, Englefood Cliffs, NJ,1994.

54 D.Framer and E.H.Spafford.The COPS security checker system.In *the Summer Usenix Conference*, Anaheim,CA,1990.

55 S.Garfinkel and G.Spafford, *Practical Unix & Internet Security*. O'Reilly & Associates Sebastopol, CA, 2nd edition, 1996.

56 M.Gasser, *Building a Secure Computer System*. Van Nostrand Reinhold, New York,1988.

57 M.Gasser, A.Goldstein, C.Kaufman,and B.Lampson. The digital distributed system security architecture. In *Proceedings of the 1989 National Computer Security Conference* ,1989 .

58 J.A.Goguen and J.Meseguer.*Security policies and security models. In Proceedings of the 1982 IEEE Symposium on Security and Privacy, pp 11-20, 1982.*

59 L.Gong, M.Mueller, H.Prafullchandra,and L.Schemers.Going beyond the sandbox: An overview of the new security architecture in the Java Development Kit 1.2. In *USENIX Symposium on Internet Technologies and Systems*, Menterey,CA,December 1997.

60 I.Greenberg. Distributed database security. Technical Report SRI-Report 8772, SRI Computer Science Laboratory, 1991.

61 Object Management Group.*Common Object Request Broker:Architecture and Specificatoin, Au-gust 1997.*

62 D.Grover (ed). The Protection of Computer Software - its technology and applications. Cambridge University Press,Cambridge,2nd edition,1992.

63 L.Hadfield, D.Hatter,and D.Bixler. *WindowsNT Server 4 Security Handbook*. Que Corporation, Indianaplis, IN,1997.

64 M.A.Harrison, M.L.Ruzzo,and J.D.Ullman Protection in operating systems. *Communications of the ACM*, 19(8) pages 461-471,August 1976.

65 M.E.Hellman. A cryptanalytic time -memory trade-off. *IEEE Transactions on Information Theory*, 26(4) pages 401-406, 1980.

66 J.Hennessy and D.Patterson.*Computer Architecture -A Quantitative Approach*. Morgan-Kaufmann Publishers, San Mateo, CA,1990.

67 H.J.Highland. Random bits and bytes. *Computers & Security*, 8(6) pages 469-478,1989.

68 S.Jajodia and V.Atluri. Alternative correctness criteria for concurrent execution of transactions in multi-level secure databases. In *Proceedings of the 1992 IEEE Symposium on Research in Security and Privacy*, pp 216-224,1992.

69 S.Jajodia and B.Kogan. Integrating an object-oriented data model with multi-level security. In *Proceedings of the 1990 IEEE Symposium on Research in Security and Privacy*, pp 76-85.

70 S.Jajodia and R.Sandhu. Polyinstantiation integrity in multilevel relations. In *Proceedings of the 1990 IEEE Symposium on Research in Security and Privacy*, pp 104-115,1990.

71 S.Jajodia and R.Sandhu. Polyinstantiation for cover stories. In *ESORICS'92*, pp 307-328. Springer LNCS 648, 1992

72 H.S.Javitz and A.Valdes. The SRI IDES statistical anomaly detector. In *Proceedings of the 1991 IEEE Symposium on Research in Security and Privacy*, pp 316-326,1991.

73 D.Kahn. *The codebreakers*. Macmillan Publishing Company, New York,1967.

74 M.H.Kang, A.P.Moore,and I.S.Moskowitz. Design and assurance strategy for the NRL pump. *IEEE Computer*, 31(4) pages 56-64, November 1998.
75 P.A.Karger. Implementing commercial data integrity with secure capabilities. In *Proceedings of the 1991 IEEE Symposium on Research in Security and Privacy*, pp 130-139,1991.
76 P.A.Karger, M.E.Zurko, D.W.Bonin, A.H.Mason,and C.E.Kahn. A VMM security kernel for the VAX architecture. In *Proceedings of the 1990 IEEE Symposium on Research in Security and Privacy*, pp 1-19,1990.
77 T.F.Keefe,M.B.Thuaisingham,and W.T.Tsai. Secure query processing strategies. IEEE Computer, 22(3) pages 63-70, March 1989.
78 T.F.Keefe and T.W Tsai. Multiversion concurrency control in multilevel secure database management systems. In *Proceedings of the 1990 IEEE Symposium on Research in Security and Privacy*, pp 369-383, 1990.
79 T.F.Keefe,W.T.Tsai,and J.Srivastava. Database concurrency control in multilevel secure database management systems. *IEEE Transaction on Knowledge and Data Engineering,* 5(6) pages 1039-1055, December 1993.
80 J.Kohl and C.Neumann. The Kerboros Network Authentication Service (VS), September 1993, Internet RFC 1510.
81 L.Lamport. Constructing digital signatures from a one-way function. Technical Report CLS-98, SRI International Computer Science Laboratory October 1979.
82 B.Lampson. Protection. *ACM Operating Systems Reviews,* 8,1974.
83 B.Lampson, M.Abadi, M.Burrows,and E.Wobber. Authentication in distributed systems: Theory and practice. *ACM Transactions on Computer Systems,* 10(4) pages 1039-1055, December 1992.
84 C.Landwehr. The best available technologies for computer security. *IEEE Computer,* 16(7) pages 86-100,1993.
85 C.E.Landwehr, A.R.Bull, J.P.McDermott,and W.S.Choi. A taxonomy of computer program security flaws,with examples. *ACM Computing Surveys*, 26(3), 1994.
86 J.C.Laprie. *Basic Concepts and Terminology* . Springer-Verlag, Vienna, 1992.
87 T.M.P.Lee. Using mandatory integrity to enforce 'commercial'security. In *Proceedings of the 1991 IEEE Symposium on Research in Security and Privacy*, pp 140-146,1991.
88 J.Linn. *Generic Security Service Application Program Interface*, January 1997. Internet RFC 2078.
89 R.J.Lipton and L.Snyder. On synchronization and security. In Demillo R.D, *et al*., editors *Fundamentals of Secure Computation*. Academic Press, New York, 1978.
90 T.F.Lunt. Aggregation and inference: Facts and fallacies. In *Proceedings of the 1989 IEEE Symposium on Research in Security and Privacy*, pp 102-109, 1989.
91 T.F.Lunt.D.E.Denning,R.R.Schell, M.Heckman,and W.R.Shockley. The SeaView securrity model. *IEEE Transactions on Software Engineering*, 16(6) pages 593-607,1990.
92 T.F.Lunt, A.Tamarn, F.Gilham, R.Jagganathan, C.Jalalai, H.S.Javits, A.Valdes,and P.G.Neumann. A real-time intrusion detection expert system. Technical Report SRI-CSL-90-05, SRI Computer Science Laboratory, 1990.
93 D.MacKenzie and G.Pottinger. Mathematics,technology,and trust: Formal verification, computer security, and U.S.Military. *IEEE Annals of the History of Computing*, 19(3) pages 41-59,1997.
94 D.Malkhi, M.K.Reiter,and A.D.Rubin. Secure execution of Java applets using a remote playground.In *Proceedings of the 1998 IEEE Symposium on Research in Security and Privacy*, pp 40-51, 1998.
95 G.McGraw and E.W.Felten. *Java Secrity*. Jhon Wiley & Sons, New York,1997.
96 J.McLean. Reasoning about security models. In *Proceedings of the 1987 IEEE Symposium on Research in Security and Privacy*, pp 123-131, 1987.
97 J.McLean. The specification and modeling of computer security, *IEEE Computer,* 23(1) pp 9-16, January 1990.

98 J.McLean. Security models. In J. Marciniak, editor, *Encyclopedia of Software Engineering.* John Wiley & Sons, New York, 1994.
99 A.J. Menezes, P.C. van Oorschot, and Vanstone S.A. *Handbook of Applied Cryptography.* CRC Press, Boca Raton, FL, 1997.
100 M.Milenkovic. *Operating Systems-Concepts and Design.* McGraw-Hill, New York, 2nd edition, 1992.
101 J.K. Millen and T.F. Lunt. Security for object-oriented database systems. In *Proceedings of the 1992 IEEE Symposium on Research in Security and Privacy*, pp 260-272, 1992.
102 B.F Miller, L. Frederiksen, and B. So. An empirical study of the reliability of Unix utilities. *Communications of the ACM, 33(*12) pages 32-44, December 1990.
103 S.P. Miller, B.C. Neuman, J.I. Schiller, and J.H. Saltzer.Section e.2.1:Kerberos authentication and authorization system. Technical report, MIT Project Athena, Cambridge, MA, 1987.
104 R.Morris and K.Thompson.Password security:A case history. *Communications of the* ACM,22(11) pages 594-597, November 1979
105 R.T.Morris.A *Weakness in the 4.2BSD Unix TCP/IP Software.* Bell Labs Computer Science Technical Report, February 1985.
106 T.J.Mowbray and R.Zahavi. *the Essential CORBA.* John Wiley & Sons, New York, 1997.
107 NCSC. *Trusted Network Interpretation, ('Red Book')* 1987.NCSC-TG-005, Version 1.0.
108 NCSC. *Trusted Network Interpretation Environments Guideline,* August 1990. NCSCTG-011.
109 R.M.Needham. Later developments at Cambridge: Titan, CAP, and the Cambridge Ring.*Annals of the History of Computing,* 14(4) page 57, 1992.
110 R.M.Needham and M.D. Schroeder. Using encryption for authentication in large networks of computers. *Communications of the ACM,* 21(12) pages 993-999,1978.
111 R.P.Nelson. *The 80386 Book.* Microsoft press, 1988.
112 US Department of Defense. *DoD Trusted Computer System Evaluation Criteria, (The Orange Book)* DOD 5200.28-STD, 1985
113 US Department of Defense. *Industrial Security Manual for Safeguarding Classified Information,* DOD 5200.22-M,June 1987.
114 National Bureau of Standards.*Data Encryption Standard.* U.S. Department of Commerce, NBS FIPS PUB 46, January 1997.
115 National Institute of Standards and Technology & National Security Agency,*Federal Criteria for Information Technology Security,* Version 1.0, 1992.
116 National Institute of Standards and Technology.*Digital Signature Standard* (*DSS*) U.S. Department of commerce, FIPS PUB 186, May 1994.
117 Commossopn of the European Communities. *Information Technology Standard Criteria*(*ITSEC*), 1993.
118 Commisson of the European Communities.*Information Technology Secuirty Evaluation Manual* (*ITSEM*), 1993.
119 E.I. Organick. *The Multics System: As Examination of Its Structure.* MIT Press, Cambridge, MA, 1972.
120 H.K.Orman. *the OAKLEY KEY Determination protocol.* Internet Draft, draft-ietf-ipsef- oakley-o2.tx.
121 R.Otte, P.Patrick, and M.Roy. *Understanding CORBA.* Prentice-Hall, Upper Saddler River,NJ, 1996.
122 J.S.Park. *AS/400 Security in a Client Server Environmant*. John Willey & Sons, New York, 1995
123 T.Parker and D.Pinkas. *Sesame v4-Overview*, Issue1. December 1995.
124 B.Pfaffenberger.*Protect Your in the Internet.* Joth Wiley & Sons,New York,1997.
125 C.P.Pfleeger.*Security in Computing.* Prentice-Hall, Englewood Cliffs,NJ,2nd edition, 1997.

126 M.K.Reiter,K.P.Birman,and R.van Renesse. A security architecture for fault tolerant systems. *ACM Transactions on Computer Systems,* 12(4) pages 340-371, 1994.
127 R.Rivest, A.Shamir,and L.Adleman. A method for obtaining digital signatures and public-key cryptosystems.*Communications of the ACM,* 21(2) pages 120-126,1978.
128 A.D.Rubin,D.Geer,and M.J.Ranum. *Web Security Sourcebook.*John Wiley & Sons, New York, 1997.
129 J.Rushby and B. Randell. A distributed secure system. *IEEE Computer,*16(7) pages 55-67, July 1983.
130 D.Russel and G.T. Gangemi Sr. *Computer Security Basics.* O'Reilly & Associates, Sebastopol, CA, 1991.
131 J.H.Saltzer. Protection and the control of information sharing in Multics. *Communications of the ACM*,17 pages 388-402, 1974.
132 S. Samalin. *Secure Unix.*McGraw-Hill, 1997.
133 R.S. Sandhu, Lattice-based access control models. *IEEE Computer,*26(11) pages 9-19, November 1993.
134 R.S.Sandhu, E.J.Coyne,H.L.Feinstein, and C.E. Youman. Role-based access control models. *IEEE Conputer,* 29(2) pages 38-47, February 1996.
135 UK ITSEC Scheme. UKSP 01 *Description of the Scheme,* March 1991.
136 UK ITSEC Scheme. UKSP 06 *UK Certified Products List*, 1998.
137 B. Schneier.*Applied Cryptography*. John Wiley & Sons, New York, 2nd edition, 1996.
138 M. Schroeder and J.H Saltzer. An hardware architecture for implementing protection rings. *Communications of the ACM,*15 pages 157-170, 1972.
139 T. Sheldon. *Windows NT Security Handbook.* McGraw-Hill, 1997.
140 T. Shimomura. *Takedown.* Martin Secker & Warburg Ltd..., 1995.
141 O. Sibert, P.A. Porras, and R. Lindel. An analysis of the intel 80x86 processor architecture and implementation. *IEEE Transaction on Software Engineering,*pp 181-199, March 1995.
142 O. Sibert, P.A. Porras, and R. Lindel. The intel 80x86 processor architecture: pitfalls for secure systems. In *Proceedings of the 1995 IEEE Symposium on Security and Privacy,*pp 211-222, 1995.
143 H.J. Smith. Privacy Policies and practices: Inside the organizational maze. *Communications of the ACM,* 36(12) pages 104-122, December , 1993.
144 K.P. Smith, B.T. Blaustein, S. Jajodia, and L. Notargiacomo. Correctness criteria for multilevel secure transactions. *IEEE Transactions on Knowledge and Data Engineering,* 8(1) pages 32-45, February 1996.
145 M. Smith. *Commonsense Computer Security*. McGraw-Hill, London, 1993.
146 E.H. Spafford. Crisis and aftermath. *Communications of the ACM.*32(6) pages 678-687, June 1989.
147 D.F. Sterne. On the buzzword 'Security Policy'. In *Proceedings of the 1991 IEEE Symposium on Research in Security and Privacy,* pp 219-230, 1991.
148 C. Stoll. *The Cuckoo's Egg.* Simon & Schuster.1989.
149 S.A. Sutton. *Window NT security Guide.* Addison Wesley Developers Press, Reading, MA, 1996.
150 Canadian System Securiy Centre. *The Canadian Trusted Computer Product Evaluation criteria*, Version 3.0e,1993.
151 J.J. Tardo and K. Alagappan. SPX _ global authentication using public-key certificates. In *Proceedings of the 1991 IEEE Symposium on Research in Security and Privacy,* pp 232-244, 1991.
152 R.K. Thomas and R.S. Sandhu. A kernelized architecture for multilevel secure object-oriented databases supporting write-up. *Journal of Computer Security.*2(2,3) pages 231-275, February 1993.

153 D. Thomson. The Sidewinder challenge-results so far.*CIPHER*, Electronic Issue 6,May 30 1995.
154 J.von Neumann.Firstdraft of a report on the EDVAC, m.d. Godfrey(ed.) .*Annals of the History of Computing,* 15(4) pages 27-75, 1993.
155 R.Wahbe, S.Lucco, T.E. Anderson, and S.L. Graham. Efficient software-based fault isolation. In *Proceedings of the Symposium on Operation System Principles,*1993.
156 W.Ware.Security control for computer systems. Technical Report R-609-1, Rand Corp Tech Report, October 1979.
157 C.Weissman.BLACKER:Security for the DDN,of A1 security engineering trades. In *Proceedings of the 1992 IEEE Symposium on Research in security and Privacy,* pp 286-292, 1992.
158 M.V.Wilkes.Time-*Sharing Computer Systems.* American Elservier,. New Yok, 1968.
159 E.Wobber, M.Abadi, M.Burrows, and B. Lampson.Authentication in the TAOS operating systems *ACM Transactions on Computer systems,* 12(1) pages 3-32, February 1994.
160 T.Y.C.Woo and S.S.Lam. Authentication for distributed systems. *IEEE Computer,* 25(1), pages 39-52, January 1992.
161 A.W.Wood, S.R.Lewis, and S.R. Wiseman. the SWORD Multilevel Secure DBMS. Technical Report RSRE 92005, DRA, Malvern, 1992.
162 P.H.Wood and S.G.kochan. *UNIX System Security.* Hayden Books, 1988.
163 Ernst & Young. *Logical Access Control.* McGraw-Hill, London, 1993.

색 인

ㄹ

ㅁ

ㅂ

ㅅ

ㅈ

ㅊ

ㅋ

ㅌ

ㅍ

ㅎ

ㅇ